TUGAS AKHIR – KS141501

OPTIMASI PENJADWALAN STAF RUMAH SAKIT DENGAN MENGGUNAKAN ALGORITMA *TABU SEARCH BASED HYPER-HEURISTICS* (STUDI KASUS: RUMAH SAKIT IBU DAN ANAK KENDANGSARI)

HOSPITAL STAFF ROSTERING OPTIMIZATION USING TABU SEARCH BASED HYPER-HEURISTICS ALGORITHM (STUDY CASE: IBU DAN ANAK HOSPITAL KENDANGSARI)

ZULI MAULIDATI
NRP 5214100701

Dosen Pembimbing
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR – KS141501

OPTIMASI PENJADWALAN STAF RUMAH SAKIT DENGAN MENGGUNAKAN ALGORITMA TABU SEARCH BASED HYPER-HEURISTICS (STUDI KASUS: RUMAH SAKIT IBU DAN ANAK KENDANGSARI)

ZULI MAULIDATI
NRP 5214100701

Dosen Pembimbing
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT – KS141501

HOSPITAL STAFF ROSTERING OPTIMIZATION USING TABU SEARCH BASED HYPER-HEURISTICS ALGORITHM (STUDY CASE: IBU DAN ANAK HOSPITAL KENDANGSARI)

ZULI MAULIDATI
NRP 5214100701

Supervisors
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

INFORMATION SYSTEMS DEPARTMENT
Information and Communication Technology Faculty
Sepuluh Nopember Institut of Technology
Surabaya 2018

LEMBAR PENGESAHAN

OPTIMASI PENJADWALAN STAF RUMAH SAKIT DENGAN MENGGUNAKAN ALGORITMA TABU SEARCH BASED HYPER-HEURISTICS (STUDI KASUS: RUMAH SAKIT IBU DAN ANAK KENDANGSARI)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

ZULI MAULIDATI
NRP. 5214100701

Surabaya, Januari 2018

Plh KEPALA
DEPARTEMEN SISTEM INFORMASI

Edwin Riksakomara, S.Kom., M.T
NIP 196907252003121001

LEMBAR PERSETUJUAN

OPTIMASI PENJADWALAN STAF RUMAH SAKIT DENGAN MENGGUNAKAN ALGORITMA TABU SEARCH BASED HYPER-HEURISTICS (STUDI KASUS: RUMAH SAKIT IBU DAN ANAK KENDANGSARI)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

ZULI MAULIDATI
NRP. 5214100701

Disetujui Tim Penguji : Tanggal Ujian : 9 Januari 2018
Periode Wisuda : Maret 2018

Ahmad Muklason, S.Kom., M.Sc., Ph.D. **(Pembimbing I)**

Wiwik Anggraeni, S.Si, M.Kom **(Penguji I)**

Faizal Mahananto, S.Kom, M.Eng., Ph.D. **(Penguji II)**

OPTIMASI PENJADWALAN STAF RUMAH SAKIT DENGAN MENGGUNAKAN ALGORITMA *TABU SEARCH BASED HYPER-HEURISTICS* (STUDI KASUS: RUMAH SAKIT IBU DAN ANAK KENDANGSARI)

Nama Mahasisiwa	**:**	**Zuli Maulidati**
NRP	**:**	**05211440007001**
Departemen	**:**	**Sistem Infromasi**
Dosen Pembimbing	**:**	**Ahmad Muklason, S.Kom., M.Sc Ph.D**

ABSTRAK

Penjadwalan staf rumah sakit atau dikenal sebagai staff healthcare rostering merupakan permasalahan kompleks yang harus dihadapi setiap rumah sakit. Rumah sakit harus mempertimbangkan banyak aspek seperti jumlah perawat, pembagian shift, cost, kesempatan libur atau cuti dan constraint yang lain. Karena banyak pertimbangan tersebut, penjadwalan secara manual akan menjadi sangat sulit dan tidak bisa memberikan solusi yang optimal. Maka perlu adanya suatu model matematis untuk memudahkan permasalahan penjadwalan dengan menemukan solusi yang paling optimal. Permasalahan tersebut lebih dikenal dengan istilah nurse rostering problem (NRP). Secara umum pemodelan nurse rostering atau heathcare staff rostering harus memperhatikan batasan-batasan objek sehinga dapat menghasilkan hasil yang optimal. Masalah lain dari penjadwalan staf merupakan masalah keadilan antar staf yang bertugas, bagaimana pembagian alokasi libur, waktu kerja, atau tempat tugas menjadi dimensi yang perlu dipertimbangkan. Banyak Penelitian menyebutkan bahwa indeks kepuasan suatu perawat atau staf dalam bekerja dipengaruhi oleh tingkat keadilan dalam pembagian jadwal. Untuk menyelesiakan permasalahan tersebut, pada penelitian ini akan dilakukan penjadwalan

dengan menggunakan algoritma tabu search hyperheuristic. Algoritma Tabu Search hyper-heuristic akan digunakan memberikan solusi terhadap masukan permasalahan dengan cara menghasilkan heuristik baru dengan menggunakan heuristic yang sudah ada. Hasil optimasi penjadwalan dengan algoritma tabu search hyper heuristics pada Penelitian ini dapat diterima. Semua hard constraint pada setiap unit dapat terpenuhi dan soft contrsaint yaitu nilai jains fairness pada masing-masing unit setelah optimasi dibandingkan hasil jadwal otomatis meningkat mendekati nilai keadilan total yaitu satu. Nilai JFI pada unit Farmasi meningkat sebesar 47%, unit Nicu & Ruang Bayi meningkat sebesar 48%, unit IGD meningkat sebesar 20%, unit SIM & RM meningkat sebesar 2%, unit Gizi & Café meningkat sebesar 23% dan Ruang Operasi meningkat sebesar 2%.

***Kata kunci: penjadwalan staf rumah sakit, manajemen sains, framwork hyper-heuristic, algoritma tabu search, tabu search hyper-heuristic, jain fairness index**.*

HOSPITAL STAFF ROSTERING OPTIMIZATION USING TABU SEARCH BASED HYPER-HEURISTICS ALGORITHM (STUDY CASE: IBU DAN ANAK HOSPITAL KENDANGSARI)

Name	**:**	**Zuli Maulidati**
NRP	**:**	**05211440007001**
Departement	**:**	**Sistem Infromasi**
Supervisior	**:**	**Ahmad Muklason, S.Kom., M.Sc Ph.D**

ABSTRACT

Scheduling labor hospital or known as staff healthcare rostering is the complex problems that must be faced by each hospital. Hospitals must consider many aspects such as the number of nurses, the division of shifts, the cost, the chance of a holiday or leave of absence and other constraints. Because a lot of these considerations, the scheduling manually will be very difficult and can not give the optimal solution. It is necessary the existence of a mathematical model to facilitate the scheduling problems with finding the most optimal solution. The problem is known with the term nurse rostering problem (NRP). In general, the modeling of the nurse rostering or healthcare staff rostering should pay attention to the boundaries of the object so that it can produce optimal results. Another problem of staff scheduling is a matter of justice between the staff on duty, how the division of the allocation of holidays, working time, or place of duty be the dimensions that need to be considered. Many research mention that the satisfaction a nurse or staff in the work influenced by the level of fairness in the division of the schedule. To resolve these problems, this research will be done scheduling using tabu search algorithm hyperheuristic. Algorithm Tabu Search hyper-heuristic will be used to provide solutions to the input problems with how to generate a new heuristic using the heuristic that already exists.

The results of the optimization of the scheduling with algorithm tabu search hyper heuristics on the research can be accepted. All hard constraints on each unit can be met and soft contrsaint i.e. the value of the jains fairness on each unit after optimization compared to the results of the schedule of automatic increases approaching the value of justice, a total that one. The value of JFI on the Pharmaceutical unit increased by 47%, Nicu, & Baby Room increased by 48%, the unit of the Emergency room increased by 20%, SIM & RM units increased by 2%, unit of Nutrition & Café increased by 23% and the Operating Room increased by 2%.

Keywords: staff scheduling hospital, management science, framwork hyper-heuristic, tabu search algorithm, tabu search hyper-heuristic, jain fairness index.

KATA PENGANTAR

Bismillahirohmanirrohim

Puji Syukur Kepada Allah SWT karena atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan buku tugas akhir dengan judul
"**OPTIMASI PENJADWALAN STAF RUMAH SAKIT DENGAN MENGGUNAKAN ALGORITMA *TABU SEARCH BASED HYPER-HEURISTICS* (STUDI KASUS: RUMAH SAKIT IBU DAN ANAK KENDANGSARI)**"
yang merupakan satu syarat kelulusan pada Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya.
Selama masa pengerjaan Tugas Akhir ini, penulis telah memperoleh banyak bantuan, bimbingan dan petunjuk dari berbagai pihak. Maka dari itu, dalam kesempatan ini penulis akan menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

- Allah SWT, yang telah memberikan kesehatan, kemudahan, kelancaran, dan kesempatan untuk penulis hingga dapat menyelesaikan Tugas Akhir ini.
- Kedua orangtua, adik, kakak, dan seluruh keluarga yang selalu hadir dan senantiasa mendoakan dan memberikan kasih sayang serta semangat tiada henti untuk menyelesaikan Tugas Akhir ini
- Bapak Dr. Ir. Aris Tjahyanto, M.Kom, selaku Ketua Departemen Sistem Informasi ITS, yang telah menyediakan fasilitas terbaik untuk kebutuhan penelitian mahasiswa.
- Bapak Ahmad Muklason, S.Kom., M.Sc., Ph.D. Selaku Dosen Pembimbing yang telah banyak meluangkan waktu untuk membimbing mengarahkan dann mendukung dalam penyelesaian Tugas Akhir
- Bapak Tony Dwi Susanto, ST, MT, Ph.D. selaku dosen wali yang telah memberikan arahan terkait perkuliahan di Departemen Sistem Informasi

- Seluruh dosen pengajar beserta staf di Departemen Sistem Informasi, FTIK ITS Surabaya yang telah memberikan ilmu dan bantuan kepada penulis selama 7 semester ini
- Teman-teman seperjuangan laboratorium RDIB yang selalu memberikan dukungan positif hingga penulis bisa menyelesaikan Tugas Akhir ini khusunya opor, nita fata dan fahrur
- Teman-teman *Trainer* Keilmiahan ITS INTEGRATOR, OSIRIS, BEM FTIF, Asisten WKTI, Pendamping Keilmiahan ITS, Racana Sepuluh Nopember yang selalu memberikan dukungan kepada penulis.
- Sahabat sahaba angkatan D14 yang selalu mendukung dan memberikan semanagat positif, memberikan canda tawa, susah duka untuk penulis dalam menjalani perkuliahan ini
- Keluarga Besar CSSMoRA ITS beserta pembina yang selalu ada memberikan pengayoman, kenyamanan, dan dukungan kepada penulis selama perkuliahan.
- Teman-teman Kontrakan MANJA Ocha, Nani, Icha, dan Umdah, yang telah menemani penulis baik masa sulitt dan masa senang selama 2 tahun terakhir dalam masa perkuliahan penulis
- Teman-teman *Teman Masa Gitu* yang selalu memberikan dukungan kepada penulis dalam mejalani perkuliahan hingga masa pengerjan tugas akhir.
- Sahabat saya yang selalu mendukung, mendoakan dan membantu dalam proses pengerjaan tugas akhir ini sehingga dapat terselesaikan dengan baik

Penulis menyadari bahwa Tugas Akhir ini masih belum sempurna dan memiliki banyak kekurangan di dalamnya. Dan oleh karena itu, penulis meminta maaf atas segala kesalahan yang dibuat penulis dalam buku Tugas Akhir ini. Penulis membuka pintu selebar-lebarnya kepada pihak-pihak yang ingin memberikan kritik, saran, masukan, dan penelitian selanjutnya yang ingin menyempurnakan karya, dan Tugas Akhir ini. Semoga buku Tugas Akhir ini bermanfaat bagi seluruh pembaca.

Surabaya, 5 Januari 2018

Zuli Maulidati

Halaman ini sengaja dikosongkan

DAFTAR ISI

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

Dalam bab ini akan dijelaskan gambaran umum mengenai tugas akhir yang diangkat. Hal tersebut meliputi latar belakang masalah, perumusan masalah, batasan tugas akhir, tujuan tugas akhir, dan relevan atau manfaat kegiatan tugas akhir. Selain itu, akan dijelaskan pula relevansi tugas akhir dengan laboratorium penelitian penulis.

1.1. Latar Belakang

Rumah Sakit Ibu dan Anak Kendangsari Surabaya merupakan rumah sakit yang dirancang untuk memberikan layanan kebutuhan ibu dan anak. Rumah sakit ini merupakan rumah sakit swasta yang didirikan oleh yayasan PT. Sandra Buana Medika pada tanggal 4 April 2009. Rumah Sakit Ibu dan Anak Kendangsari Surabaya menjalanakan fungsinya dengan memberikan pelayanan kesehatan prima bagi ibu dan anak yang didukung dengan berbagai program unggulan seperti senam hamil, instalasi menyusui dini, laboratorium USG, potong rambut bayi, dan pijat bayi. Rumah sakit ibu dan anak kendangsari juga memiliki berbagai fasilitas rumah sakit seperti poli anak, poli olygon, fasilitas rawat jalan fasilitas rawat inap dan lain sebaginya. Agar layanan tersebut dapat berjalan dengan baik maka pihak rumah sakit harus dapat mengelolah segala sumber daya secara efektif dan efisien serta dapat meminimalkan anggaran belanja rumah sakit.

Salah satu langkah pengelolahan sumber daya rumah sakit yaitu dengan cara melakukan penjadwalan staf rumah sakit baik kategori medis maupun non medis. Dalam organisasi kesahatan penjadwalan staf rumah sakit atau *healtcare staff rostering* merupakan masalah yang cukup sulit untuk dipecahkan. Hal tersebut dapat dipengaruhi oleh jumlah pasien yang tidak terkendali, berbagai macam penyakit yang diderita pasien serta budaya yang melekat pada pegawai rumah sakit seperti jam kerja, shift libur dan cuti kerja. Faktor lain yang menjadi

permasalahan *healthcare staff rostering* adalah sulitnya menentukan berapa panajang dalam satu shift dan pembagian shift tersebut. Dengan banyaknya kasus dan faktor-faktor yang mempengaruhinya, kasus penjadwalan tenaga rumah sakit juga dapat diselesiakan dengan berbagai metode yang berbeda-beda.

Penjadwalan staf rumah sakit atau *healthvare staff rostering* merupakan salah satu permasalahn penting dan sulit yang dihadapi oleh setiap rumah sakit. Penjadwlan staf harus dilakukan dengan mempertimbangkan banyak aspek seperti jumlah staf, pembagian shift, *cost*, kesempatan libur atau cuti dan *constraint* yang lain. Aspek aspek tersebut sangat mempengaruhi bagaimana membuat penjadwalan secara optimal. Kebanyakan penjadwalan banyak dilakukan secara manual, hal itu menimbulkan banyak kasus dan kelemahan pemerataan pembagian tugas setiap staf yang bertugas. Sehingga beberapa pegawai atau staf mengalami penurunan performa kerja atau terdapat faktor ketidakpuasan dan tidak bahagia(*unhappy*). Oleh karena itu dibutuhkan suatu model matematis yang dapat digunakan untuk melakukan penjadwalan secara otomatis. konsep optimasi yang dapat memberikan hasil yang baik harus mempertimbangkan setiap variabel yang mempengaruhi fungsi tujuan dan *contrains-constraint* yang ada baik *hard constraint* atau *soft constraint*[1].

Penelitian optimasi penjadwalan merupakan bukan hal baru dalam bidang riset operasi atau manajemen sains. Penelitian tersebut sudah banyak dilakukan di luar negeri dengan menggunakan berbagai pilihan metodologi dan berbagai model penyelesainya. Penelitian tersebut biasanya dikerjakan dengan menggunakan metodologi yang cocok untuk *timetabling* salah satunya yaitu *tabu search*. Algoritma *tabu search* merupakan salah satu framework *meta-heuristic* yang melakukan optimasi *timetabling* atau penjadwalan[2].

Algoritma *tabu search* adalah sebuah metode optimasi yang berbasis pada *local search*[3]. Algoritma *tabu search*

digunakan untuk mencari solusi yang bergerak ke solusi berikutnya dengan cara memilih solusi yang terdekat dan tidak dilarang atau tabu[3]. Algoritma *tabu search* juga banyak digunakan kasus penelitian seperti melakukan penjadwaan, pemilihan rute terbaik, tata letak tempat dan lain sebagainya.

Beberapa penelitian terdahulu menemukan *algoritma tabu search hyper-heuristic* dapat memberikan solusi yang lebih baik dari agloritma *tabu search based meta-heuristic*. Dimana algoritma *hyper-heuristic* dibangun dengan algoritma generik dapat digunakan di bebrapa kasus yang berbeda sedangkan *metaheuristic* hanya dapat digunakan pada satu kasus tertentu saja, ada saatnya algoritma *metaheuristic* dapat menghasilkan solusi yang sangat optimal pada satu kasus akan tetapi sangat buruk pada kasus yang lain. Hasil dari penelitian tersebut mengatakan, dengan menggunakan algoritma *tabu search based hyper-heuristic* dibadingkan dengan investigasi yang dilakukan *learning mechanism* algoritma *tabu search based hyperheuristic* dapat memberikan hasil yang lebih efektif dan lebih cepat[4].

Untuk itu, di dalam tugas akhir ini akan diusulkan sebuah algoritma matematik untuk menyelesaikan permasalahan penjadwalan staff tenaga medis dan non medis di Rumah Sakit Ibu dan Anak Kendangsari Surabaya yaitu algoritma *tabu search based hyper-heuristic*. Penerapan gabungan *tabu search heuristic* dalam penelitian ini bertujuan untuk mencari nilai paling optimal di setiap kali dilakukan iterasi pemilihan. *Tabu search* akan digunakan sebagai *tools* untuk memilih *heuristic* baru pada *low level* dengan menggunkan *heuristic* lama. Hasil *Heuristic*s yang lebih baik akan digunakan iterasi kembali sedangkan *heuristic* yang buruk akan dimasukan ke dalam *memory tabu list* agar tidak dapat dipilih kembali. Sehinga hasil yang diharapkan pada penelitian ini adalah dapat menghasilkan algoritma penjadwalan otomatis yang dapat membantu mengoptimalkan penjadwalan staf tenaga medis dan non medis di Rumah Sakit Kendangsari Surabaya.

1.2. Perumusan Masalah

Berdasarkan uraian latar belakang yang telah dijelaskan, maka rumusan masalah dari tugas akhir ini, yaitu:

1.2.1. Bagaimana model matematis permasalahan penjadwalan staf di Rumah Sakit Ibu dan Anak kendangsari Surabaya?
1.2.2. Bagaimana menjadwalkan staf rumah sakit dengan menggunakan algoritma *tabu search based hyper-heuristics*?
1.2.3. Bagaimana perbandingan hasil penjadwalan staf rumah sakit kondisi saat ini dengan penjadwalan otomatis?

1.3. Batasan Pengerjaan Tugas Akhir

Dari permasalahan yang disebutkan pada perumusan masalah diatas, batasan pemasalahan dalam tugas akhir ini adalah:

1.3.1. Tugas Akhir ini dilakukan di Rumah Sakit Ibu dan Anak Kendangsari Surabaya
1.3.2. Data yang digunakan sebagai *hard constraint* adalah aturan penjadwalan yang diterapkan di Rumah Sakit Ibu dan Anak kendangsari Surabaya
1.3.3. Data yang digunakan sebagai *soft constraint* adalah ukuran optimaliti yang ditetapkan pada Rumah Sakit Ibu dan Anak kendangsari Surabaya, ukuran optimaliti merupakan ukuran keadilan antar staf dalam memproleh shift libur.
1.3.4. Hasil dari tugas akhir ini adalah sebuah *algoritma* penjadwalan otomatis yang dibangun dengan menggunakan bahasa pemograman java berbasis lokal.

1.4. Tujuan Tugas Akhir

Tujuan yang hendak dicapai dalam pengerjaan tugas akhir ini adalah:

1.4.1. Mendapatkan model matematis yang berasal dari atuaran penjadwalan staf di Rumah Sakit Ibu dan Anak kendangsari Surabaya

1.4.2. Mengoptimalkan penjadwalan staf di Rumah Sakit Ibu dan Anak kendangsari surabaya dengan menggunakan *tabu search algoritma hyper-heuristic* sesuai dengan batasan batasan yang berlaku sesuai dan dengan model yang dirumuskan

1.4.3. Mendapakan *Algoritma* penjadwalan yang optimal untuk penjadwalan staf di Rumah Sakit Ibu dan Anak Kendangsari Surabaya.

1.5. Manfaat Tugas Akhir

Manfaat yang diberikan dengan adanya tugas akhir ini adalah sebagai berikut:

1.5.1. Manfaat Teoritis

Adanya penelitian ini diharapkan dapat menambah khazanah keilmuan dalam bidang sistem informasi khususnya mengenai implementasi algoritma *tabu search hyper heuristics* pada kasus penjadwalan staf rumah sakit.

1.5.2. Manfaat Praktis

Bagi Pihak Rumah Sakit Ibu dan Anak Kendangsari Surabaya Tugas akhir ini diharapkan dapat membantu pihak Rumah Sakit Ibu dan Anak Kendangsari dalam mengoptimalkan penjadawalan staf rumah sakit sehingga dapat meningkatkan pelayanan yang lebih efektif dan efisien.

1.5.3. Bagi Mahasiswa

Tugas akhir ini diharapkan dapat membantu mahasiswa untuk menambah wawasan dan pengetahuan terkait dengan topik penjadwalan staf rumah sakit.

1.5.4. Bagi Penelitian Berikutnya

Tugas akhir ini diharapkan dapat memberikan motivasi dan gambaran bagi peneliti lain untuk melakukan pengembangan terhadap studi mengenai penjadwalan staf rumah sakit.

1.6. Relevansi

Berikut merupakan *table roadmap* penelitian laboratorium Rekayasa Data dan Intelegensi Bisnis Departemen Sitem Infomasi ITS:

Tabel 1.1 Road Map Penelitian RDIB

Rekayasa Data dan Intelegensi Bisnis **Road Map Research**				
Computerized Decision Support	Data Management	Business Analytic	Knowledge Management	Intelligent Systems
▪ Decision Support System ▪ System Modelling and Analysis	▪ Database and Database Management System (DBMS) ▪ Extraction, Transformation, and Load (ETL) System ▪ Data Warehouse (DW), real-time DW, and Data Mart	▪ **Optimizations** ▪ Data/Web/Text Mining ▪ Web Analytic ▪ Peramalan	▪ Knowledge Management System ▪ Expert Locating System ▪ Ontology	▪ Expert System ▪ Artificial Neural Network ▪ Fuzzy Logic ▪ Genetic Algorithm ▪ Intelligent Agent ▪ Automated Decision System.

Berdasarkan Tabel 1.1 di atas topik tugas akhir ini menjawab pokok penelitian *Bisnis Analytics* bagaian optimasi. Hal tersebut sesuai dengan bidang ilmu riset operasi yang menjadi cakupan *research roadmap* pada laboratorium Rekayasa Data dan Inteligensi Bisnis (RDIB) yang memiliki konsentrasi penelitian terkait pemanfaatan data yang mendukung analisis bisnis dan organisasi serta pengetahuan sehingga berguna dalam pengambilan keputusan.

BAB II
TINJAUAN PUSTAKA

Dalam Bab ini, akan dijelaskan mengenai penelitian terdahulu dan landasan teori yang digunakan sebagai acuan dalam pengerjaan tugas akhir. Penelitian terdahulu merupakan suatu penelitian yang pernah dilakukan oleh peneliti-peneliti sebelumnya yang diguanakan sebagai acuan tugas akhir. Landasan teori merupakan teori-teori yang berhubungan dengan pengerjaan tugas akhir.

2.1. Studi Sebelumnya

Pada sub bab ini akan diterangkan mengenai beberapa penelitian terdahulu yang telah dilakukan dan memiliki relevansi dengan tugas akhir ini. Penelitian terdahulu tersebut dapat dilihat pada Tabel 2.1,Tabel 2.2, Tabel 2.3, Tabel 2.4, Tabel 2.5.

Tabel 2.1 Penelitian Sebelumnya (1)

Nama Peneliti	Burke, E., dan E. Soubeiga
Tahun Penelitian	2003
Judul Penelitian	*Scheduling Nurses Using A Tabu-Search Hyperheuristic*[4]
Penjelasan Singkat	Penelitian ini membahas tentang bagaimana menyusun jadwal 30 perawat yang bekerja di Rumah Sakit umum UK. Penjadwalan yang diharapkan pada penelitian ini adalah penjadwalan yang disusun dengan mempertimbangkan bebrapa *constraint* dan regulasi yang terdapat di rumah sakit. Penelitian ini membahasa usulan meode baru yaitu *tabu search* yang didasarkan pada *framework hyper-heuristic* dimana pada penelitian sebelumya yang banyak digunakan merupakan *framework metaheuristics*. Penelitian ini

	menggembangkan teknik *hyper-heuristic* lama *simple heyper-heuristics* ke teknik *hyper-heuristic* baru *sophisticsted hyper-heuristic*.
Hasil Penelitian	Hasil dari penelitian ini dengan menggunakan algoritma *tabu search based hyper-heuristic* dapat memberikan kualitas solusi yang sama dengan pilihan fungsi *framework hyper-heuristic* dan lebih cepat setengah waktu dari *hyperheuristic* yang lama. Dibadingkan dengan investigasi yang dilakukan *learning mechanism* algoritma *tabu search based hyperheuristic* dapat memberikan hasil yang lebih efektif.

Tabel 2.2 Penelitian Sebelumnya (2)

Nama Peneliti	E.K. Burke, G. Kendall And E. Soubeiga
Tahun Penelitian	2004
Judul Penelitian	*A Tabu-Search Hyperheuristic for Timetabling and Rostering*[5]
Penjelasan Singkat	Pada penelitian in membahas tentang bagaimana algoritma *tabu-search based hyper-heuristic* memcahkan dua masalah penjadwalan yang sangat berbeda. Pada *framework hyper-heuristic* ini dilengkapi dengan model dinamik *tabu search* yang dianggap sukses dalam melakukan penelitian kasu penjadwalan perawat dengan banyak *constraint* pada rumah sakit. Ketika dibandingkan dengan algoritma genetika algoritma *tabu search hyper-heuristics* lebih memiliki performa yang *fisible*. Akan tetapi untuk *cost* algoritma

	gentika lebih optimal dibandingkan algoritma *tabu search hyper-heuristics*.
Hasil Penelitian	Hasil penelitian ini mengatakan bahwa penggunaan algoritma *tabu search hyper-heuristics* dengan membandingkan kasus yang berbeda dapat menghasilkan solusi yang baik. *Tabu search heuristic* merupakan algoritma yang umum dan mudah digunakan. *Tabu search hyperheuristic* dapat mencari permasalahan sampai ke *space low level heuristic* sesuai dengan tujuan dan kondisi yang ada.

Tabel 2.3 Penelitian Sebelumnya (3)

Nama Peneliti	Shahriar Asta, Ender ¨Ozcan, Tim Curtois
Tahun Penelitian	2016
Judul Penelitian	*A tensor based hyper-heuristic for nurse rostering*[6]
Penjelasan Singkat	Penelitian ini membahasa tentang penjadwalan perawat dengan menggunakan altoritma *tensor* berdasarkan framewok *heyper-heuristics*. Hyper-heuristik telah muncul sebagai metodologi pencarian umum yang menggabungkan dan mengatur low level heuristic sambil memecahkan masalah tersulit bidang komputasi. Di dalam penelitian ini peneliti mendeskripsikan bagaimana *hyper-heuristik* dengan menggunakan data set mampu melakukan perbaikan diri melalui analisis tensor nurse rostering.

Hasil Penelitian	Hasil yang dihasilkan dari penelitian ini adalah *hyper-heuristics* berbasis *tensor* dengan memori refresh menghasilkan solusi terbaik baru untuk empat contoh *benchmark* permasalahan penjadwalan perawat.

Tabel 2.4 Penelitian Sebelumnya (4)

Nama Peneliti	Xiang Zhong, Jingyu Zhang & Xuanqi Zhang
Tahun Penelitian	2017
Judul Penelitian	*A two-stage heuristic algorithm for the nurse scheduling problem with fairness objective on weekend workload under different shift designs*[7]
Penjelasan Singkat	Penelitian ini membahas tentang penjadwalan perawat di rumah sakit dengan contrains kesetaraan antar perawat yang bekerja di rumah sakit serta mempertimbangkan masalah perbedaan shift dan weekend. Penelitian ini dilakukan dengan menggunakan *a two stage heuristics algorthm.*
Hasil Penelitian	Hasil dari penelitian ini adalah dengan menggunakan aldoritma *a two stage heuristics algorthm* seluruh contrain yang menjadi permasalahan penjadwalan perawat dapat diterima sehingga dapat dianggap solusi optimal.

Tabel 2.5 Penelitian Sebelumnya (5)

Nama Peneliti	Muhammad Asrar Amir
Tahun Penelitian	2017

Judul Penelitian	Optimasi Penjadwalan Perawat Menggunakan Gabungan *Integer Linear Programming* Dan *Variable Neighborhood Search*. Studi Kasus Instalasi Gawat Darurat Rumah Sakit Ibnu Sina Makassar.[8]
Penjelasan Singkat	Penelitian ini membahas tentang bagaimana melakukan penjadwalan perawat di IDG Rumah Sakit Ibnu Sina Maksar dengan membandingkan dua metode yaitu *Integer Linear Programming* Dan *Variable Neighborhood Search.*
Hasil Penelitian	Hasil penelitian ini pembuatan jadwal perawat yang ada di IGD Rumah Sakit Ibnu Sina Makasar dengan mnggunakan metode *integer linear programmer* dan *variable neighborhood search* menemkan solusi yang cukup optimal. Semua *hard constraint* tidak ada yang dilanggar oleh kedua metode tersebut begitupulah *hard constraint*. Sehingaa jadwal yang dihasilkan tidak menyalahi sistemregulasi perawat yang ada di rumah sakit.

2.2. Dasar Teori

Pada sub bab ini akan dijabarkan mengenai dasar teori yang digunakan untuk mendukung pengerjaan tugas akhir ini.

2.2.1. Rumah Sakit

Menurut WHO (*World Health Organization*), rumah sakit adalah bagian integral dari suatu organisasi sosial dan kesehatan dengan fungsi menyediakan pelayanan paripurna (komprehensif), penyembuhan penyakit (kuratif) dan pencegahan penyakit (preventif) kepada masyarakat. Rumah sakit juga merupakan pusat pelatihan bagi tenaga kesehatan dan pusat penelitian medis[9].

Menurut undang undang Republik Indonesia No. 44 tahun 2009 rumah sakit adalah institusi pelayanan kesehatan bagi masyarakat dengan karakteristik tersendiri yang dipengaruhi oleh perkembangan ilmu pengetahuan kesehatan, kemajuan teknologi, dan kehidupan sosial ekonomi masyarakat yang harus tetap mampu meningkatkan pelayanan yang lebih bermutu dan terjangkau oleh masyarakat agar terwujud derajat kesehatan yang setinggi-tingginya[10].

Menurut undang-undnag resmi rumah sakit pasal satu, Rumah Sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna yang menyediakan pelayanan rawat inap, rawat jalan, dan gawat darurat[11].

2.2.2. RSIA Kendangsari Surabaya

RSIA Kendangsari MERR Surabaya adalah Rumah Sakit khusus yang dirancang unik berfokus untuk melayani kebutuhan ibu dan anak. RSIA Kendangsari MERR Surabaya didirikan perusahaan swasta PT. Sandra Buana Medikan pada tanggal 4 April 2009[12]. RSIA Kendangsari MERR Surabaya bergerak dalam lingkup bidang usaha jasa rumah sakit swasta.

RSIA Kendangsari MERR Surabaya memberikan pelayanan kesehatan yang paripurna untuk wanita dan anak. Pelayanan kesehatan diberikan secara prima dan komprehensif bagi pasien, keluarga pasien dan provider baik perusahaan maupun asuransi. Hal tersebut dibuktikan dengan banyaknya macam layanan dan fasilitas rumah sakit yang berhubungan dengan kesehatan ibu dan anak. Layanan dan fasilitas yang disediakan di RSIA diantaranya adalah layanan rawat jalan, layanan rawat inap, program kehamilan, layayanan pusat kesehatan dan tumbuh kembang anak, layanan *one day care*[12]. Selain itu RSIA juga menyediakan dokter umum dengan adanya layanan IGD yang siap selama 24 jam.

RSIA juga dilengkapi dengan layanan unggulan untuk program kehamilan dan pasca kelahiran buah hati. Seperti program senam hamil, klinik laktasi, USG, pijat bayi, kelak laktasi, dan potong rambut bayi[13]. RSIA juga menyediakan layanan yang baik dengan siapnya berbagai macam tenaga ahli seperti staf tenaga medis dan non medis, dokter spesialis anak, dokter spesialis kandungan, bidan, dokter bedah dan tenaga pendukung lainya[13].

2.2.3. Manajemen Operasi (Operation Research)

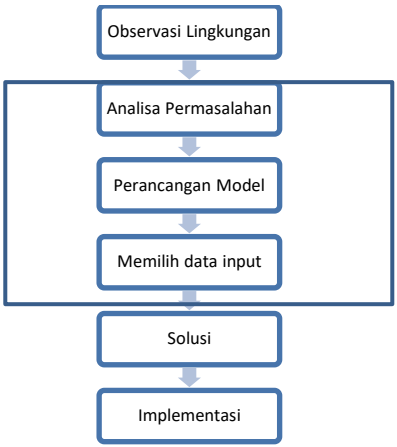
Manajemen sains merupakan ilmu yang menggunakan model matematika dalam dalam penyelasain masalah menejemen[14]. Tujuan utama dari managemen sains adalah untuk menyelesaikan masalah-masalah yang dihadapi oleh seorang menager baik yang bergerak dalam sektor publik maupun swasta dalam proses pengambilan keputusan dengan cara pendekatan model-model matematika[14].

Manajemen sains juga disebut dengan istilah riset operasi atau *operation research*. Riset operasi merupakan penerapan metode-metode ilmiah terhadap masalah-masalah rumit yang muncul dalam pengarahan dan pengelolaan dari suatu sistem besar manusia, mesin, bahan dan uang dalam industri, bisnis, pemerintahan dan pertahanan. *Operation research* atau OR merupakan disiplin ilmu yang diaplikasikan untuk membuat kepuisan yang paling baik atau optimal[15]. Istilah OR atau *operation research* pertama kali ditemukan di akhir 1930-an di inggris[15].

Penelitian operasional (OR) mencakup berbagai teknik pemecahan masalah dan metode yang diterapkan dalam pengambilan keputusan dan efisiensi yang lebih baik, seperti simulasi, optimasi matematis, teori antrian dan model proses stokastik lainnya, proses keputusan Markov, metode ekonometrik , analisis envelopment data, jaringan syaraf tiruan, sistem pakar, analisis keputusan, dan proses hirarki analitik[15]. Hampir semua teknik ini melibatkan konstruksi model

matematis yang mencoba menggambarkan sistem. Karena sifat komputasi dan statistik dari sebagian besar bidang ini, riset operasi juga memiliki ikatan yang kuat dengan ilmu komputer dan analisis.

Tahapan pengembangan O.R. juga dikenal sebagai fase dan proses O.R, memiliki enam langkah penting. Keenam langkah ini disusun dengan urutan seperti Gambar 2.1[15]



Gambar 2.1 Teknik Manajemen Sains
(Sumber: Hillier, 2001)

a. Observasi Lingkungan
   Tahap obesrvasi merupakan tahapan awal dalam teknik manajemen sains. Tahapan ini merupakan proses identifikasi permasalahan yang akan diselesaikan. Objek penelitian harus dilakukan pengamatan dan identifikasi menyeluruh secara terus menerus.
b. Analisa Permasalahan
   Tahapaan yang kedua merupakan analisa masalah. Pada tahapan ini akan dilakukan penentuan permasalahan secara jelas dan singkat.
c. Perancangan Model

Tahap yang ketiga adalah perancangan model. Model merupakan abstarksi dari permasalahan yang ada, dalam teknik manajemen sains terdapat beberapa aspek dalam pembuatan model:

1. Fungsi Tujuan
   Fungsi Tujuan merupakan tujuan yang ingin dipecahkan dalamsuatu penelitianoptimasi. Fungsi tujuan dari tugas akhir ini adalah mengoptimalkan Penjadwalan staf rumah sakit
2. Variabel Keputusan
   Variabel keputusan merupakan *variable-variable* yang belum diketahui nilainya, dan yang akan dicari nilainya dalam permasalahan yang didapatkan. Dalam tugas akhir ini didapatkan variable keputusan diambil dari penjadwalan staf RSIA Kendangsari Surabaya
3. Fungsi Batasan
   Batasan merupakan kumpulan dari beberapa fungsi yang digunakan untuk memberikan batasan variable keputusan dalam mencapai fungsi tujuan.

d. Memilih data input
   Setelah melakukan pemodelan tahapan selanjutnya adalah memasukan data observasi ke dalam model yang sudah dibangun.
e. Solusi
   Tahapan terakhir yaitu menemukan solusi dari proses optimasi yang telah dilakukan.
f. Implementasi

2.2.4. Staff Helathcare

Staf healthcare adalah orang-orang yang bertugas untuk memberikan layanan dan memperbaiki kesehatan masyarakat[9]. Menurut peraturan kemetrian kesehatan tahun 2014 sumber daya manusia rumah sakit terdiri dari a. tenaga medis; b. tenaga kefarmasian; c. tenaga keperawatan; d. tenaga kesehatan lain; e. tenaga nonkesehatan[16].

Menurut WHO *healthcare staff* merupakan praktisi kesehatan meliputi dokter, dokter gigi, ahli kebersihan gigi, apoteker, teknisi farmasi, asisten dokter, perawat, perawat praktek lanjut, perawat bedah, asisten ahli bedah, pelatih atletik, teknolog bedah, bidan, ahli diet, terapis, psikolog, ahli tulang, petugas klinik, pekerja sosial, ahli floral, terapis okupasi, dokter mata, ahli terapi fisik, radiografer, radioterapi, terapis pernafasan, audiolog, praktisi departemen operasi, teknisi medis darurat, paramedis, ilmuwan laboratorium medis, teknisi prostetik medis dan berbagai sumber daya manusia lainnya yang terlatih untuk menyediakan beberapa jenis layanan kesehatan[17].

Healthcare staff biasanya sering bekerja di rumah sakit, pusat kesehatan, dan titik layanan kesehatan, selain itu staf *haealthcare* juga dapat ditemuhi dalam pelatihan akademis, penelitian, dan administrasi. Beberapa dari mereka membuka parktek untuk menyediakan layanan perawatan dan perawatan untuk pasien di rumah-rumah pribadi[9].

2.2.5. Nurse Rostering

Kehadiran perawat dalan rumah sakit dalam 24 jam sehari merupakan suatu kebutuhan yang wajib dipenuhi. Dengan itu hampir di seluruh rumah sakit memiliki jadwal kerja setiap perawat dengan dibagi menjadi shift-shift kerja. Sebagian besar rumah sakit mengizinkan perawat untuk meminta perubahan jadwal dari jadwal yang telah ditentukan sebelumnya, sementara itu perawat yang lain dapat juga mengalami pergeseran jadwal yang diakibatkan hal tersebut. Karena hal tersebut penjadwalan perawat menjadi sesuatu yang terkesan sulit karena banyaknya batasan[1].

Constraint dalam *nurse restoring problem* dapat dikelompokkan menjadi dua kategori: (i) hubungan yang menghubungkan dua atau lebih perawat dan (ii) hal-hal yang hanya berlaku untuk satu perawat saja[6]. *Constraint* yang masuk dalam kategori pertama mencakup batasan *cover* (desakan yang harus dipertimbangkan). Ini adalah *Constraint*

yang memastikan jumlah minimum atau jumlah perawat diberikan setiap *shift* setiap hari. *Constraint* lain yang juga termasuk *constraint* kategroi pertama adalah ketrampilan yang dimiliki oleh masing-masing perawat, *contraint* yang menentukan perawat bekerja sama dengan perawat lain atau bekerja sendiri.

Batasan yang perlu dipertimbangan dalam melakukan penjadwalan dibagi menjadi dua *constraint* yaitu *hard constraint* dan *soft constraint*[1]. Berikut contoh *constraint* secara umum yang sering terjadi di kasus penelitian penjadwalan perawat:

- Kemampuan perawat bekerja
- Kebutuhan perawat di setiap shift
- Kesamaan kerja pada satu shft
- Lama bekerja dalam satu shift/hari
- Hari libur perawat
- Waktu minumum istirahat antara shift satu ke shift berikunya
- Tipe shift
- Hari akhir pekan
- Catatan kerja perawat
- Skil perawat

Berikut model matematis optimasi penjadwalann perawat (*nurse rostering problem*)[6]:

Diketahui *Variable problem nurse rorseting*:

E = perawat yang dijadwalkan, $e \in E$

T = shift yang akan dialokasikan, $t \in T$

D = hari yang dijadwalkan, $d \in \{1, \cdot \cdot \cdot |D|\}$

Re = keinginan perawat e, $r \in Re$

We = Batas beban kerja staf e, w ∈ We

Parameter:

- r_{er}^{Max} = jumlah maksimum keinginan perawat *r* yang sesuai dengan jadwal kerja perawat *e*.
- r_{er}^{Min} = jumlah maksimum keinginan perawat r yang sesuai dengan jadwal kerja perawat e.
- a^{er} = hubungan bobot kerja perawat e dengan keinginan perawat *r*.
- V_{ew}^{Max} = jumlah jam kerja maksimum yang akan diberikan kepada staf dalam jangka waktu yang ditentukan oleh batasan kerja
- V_{ew}^{Min} = jumlah jam kerja minimum yang akan diberikan kepada staf dalam jangka waktu yang ditentukan oleh batasan kerja w
- b^{ew} = bobot terkait dengan batas beban kerja w untuk kariyawan e.
- s_{td}^{Max}= jumlah maksimum shift t dalam satu hari d.
- s_{td}^{Min}= jumlah minimum shift t dalam satu hari d.
- c^{td} = Bobot masing masing tipe shift t dalam satu hari d

Membuat fungsi batasan atau *Constraints:*

$$\sum_{x_{etd}} \boldsymbol{x_{etd}} \leq \mathbf{1}, \ \forall \boldsymbol{e} \in \boldsymbol{E}, \boldsymbol{d} \in \boldsymbol{D} \qquad (2.1)$$

Merumuskan Fungsi Tujuan:

$$\boldsymbol{minf(s)} = \sum_{e \in E} \sum_{i=1}^{4} \boldsymbol{f_{e,1}(x)} + \sum_{t \in T} \sum_{d \in D} \sum_{i=5}^{6} \boldsymbol{f_{t,d,i}(x)} \qquad (2.2)$$

Where:

$$f_{e,1}(x) = \sum_{e \in R_e} \max\{0, (n_{er} - r_{er}^{max}) a_{er}\} \quad (2.3)$$

$$f_{e,2}(x) = \sum_{e \in R_e} \max\{0, (r_{er}^{min} - n_{er}) a_{er}\} \quad (2.4)$$

$$f_{e,3}(x) = \sum_{e \in W_e} \max\{0, (p_{ew} - v_{ew}^{max}) b_{ew}\} \quad (2.5)$$

$$f_{e,4}(x) = \sum_{e \in W_e} \max\{0, (v_{ew}^{min} - p_{ew}) b_{ew}\} \quad (2.6)$$

$$f_{e,5}(x) = \max\{0, (s_{td}^{min} - q_{td}) C_{td}\} \quad (2.7)$$

$$f_{e,6}(x) = \max\{0, (q_{td} - s_{td}^{max}) C_{td}\} \quad (2.8)$$

2.2.6. Nilai Jain Fairness Index

Beberapa penelitian menunjukkan bahwa *rosering* staf atau perawat pada rumah sakit memiliki tingkat kualitas yang tinggi disebabkan oleh tingkat kepuasan stafnya[18]. Hal tersebut merupakah hal yang dianggap penting, karena dengan meningkatkan Kepuasan kerja staf atau perawat, tingkat retensi mereka juga cenderung meningkat. Faktor yang mempengaruhi kualitas penjadwalan staf pada rumah sakit merupakan sebuah daftar termasuk jam kerja, apakah perawat dapat bekerja dalam jumlah minimum jam kerja per hari selama berturut-turut. Distribusi pekerjaan yang adil juga dapat mempengaruhi tingkat kepuasan staf bekerja[18]. Untuk itu dalam pendekatan ini penulis menyajikan bagaimana membuat jadwal staf yang terdistridibusi secara adil.

Pada Penelitianya, Wanner menyebutkan bahwa *fairness* pada penjadwalan staf dianggap sebagai ukuran kualitas penjadwalan[19]. Warner bahkan menunjukkan distribusi keadilan pekerja memiliki banyak keuntungan dibandingkan dengan pola jadwal yang berulang ulang. Burke dkk[4] juga mengatakan bahwa sebagian besar Penelitian penjadwalan

perawat tidak secara eksplisit membahas tentang keadilan, padahal faktor *fairness* mrupakan faktor penyeimbang pembagian tugas antara staf[19].

Rajendra jain menemukan sebuah model matematis yang dinamakan *Jain fairness index* untuk menghitung tingkat *fairness*[20]. Dengan mengalokasikan sumber daya pada jadwal, indeks keadilan adalah angka nyata yang dapat mengukur seberapa adil atau tidak adil sumber daya dibagi di antara slot jadwal[21]. *Jain fairness index* dapat dihitung dengan persamaan matematis berikut.

$$JFI = \frac{(\sum_{i=1}^{i} x_i)^2}{(n * \sum_{i-1}^{i} (x_i)^2)}$$

Dimana nilai x_i merupakan nilai variabel yang akan dibandingkan keadilannya dan n merupakan jumlah slot yang dialokasikan dengan x_i.

Nilai *Jain Fairness Index* dibatasi antara range 0 sampai 1, dimana jika nilai JFI lebih besar maka menyiratkan solusi yang diusulkan semakin adil begitupulah sebaliknya jika semakin mendekati 0 maka tingkat keadilan semakin kecil[22]. Sehingga nilai JFI 1 dianggap sebagai nilia keadilan total[22].

2.2.7. Framework Hyper-Heuristiscs

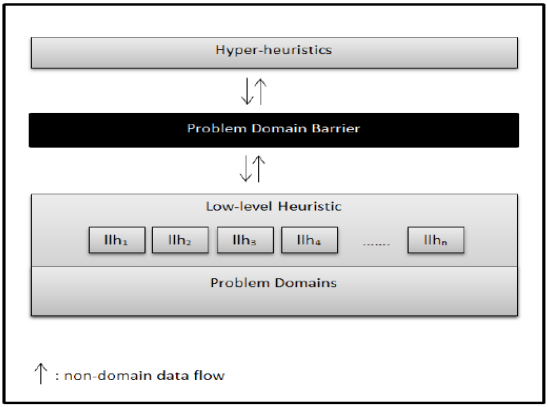
Hyper-heuristic adalah metode pencarian *heuristic* yang mengotomisasi dengan menggabungkan teknik pembelajaran mesin, proses pemilihan, penggabungan, dan menghasilkan adaptasi *heuritics* baru untuk menyelesaikan pencarian komputasi secara efisien. Tujuanya adalah memilih dan mengkombinasikan *heuristics* yang sederhana menghasilakan *heuristic* baru dengan komponen *heuristic* yang sudah ada[23].

Ide utama dari *hyper heuristic* adalah pengembangan algoritma yang lebih generik dan bertentangan dengan pendekatan *meta heuristic.* Pada umumnya kelemahan *meta-heuristics* adalah

algoritma memerlukan *parameter tunning* yang intensif dan memerlukan problem domain yang spesifik. Sehingaa dalam kasus yang berbeda diperlukan adanya *parameter tunning* yang berbeda pula. Hal ini mengakibatkan adanya pengaruh terhadap perfoma algoritma. Dalam hal ini satu *problem instance* dapat memiliki performa algoritma yang sangat baik sedangkan *problem instance* yang lain akan memiliki performa algoritma yang sangat buruk[24].

Framework *hyper-heuristics* secara umum dapat dibangun dengan algoritma seperti *pseudocode* berikut ini[25]:

Step 1. Construct initial solution
Step 2. Do
Consider heuristics that are not tabu.
Apply chosen heuristic and make the heuristic tabu.
Update Solution.
Until terminating condition



Gambar 2.2 Framework Hyper-Heuristics
(Sumber: Muklason, 2017)

Berdasarkan Gambar 2.2 bagian inti dari *framework Hyper heuristic* merupakan *domain barier*. Hal itu mengakibatkan

hyper heuristic tidak memiliki informasi domain pada *low level*[24]. *Hyper heuristic* mampu bekerja pada permasalahan *high level* pada ruang pencarian yang *low level*[24]. Algoritma *hyper-heuristic* akan bekerja diatas *search space* berupa *lower-level heuristics* sedangkan *meta-heuristic* akan bekerja diatas search space berupa *solution space* atau *low level heuristic*. Sehingga *hyper-heuristic* tidak secara langsung bergantung pada problem domain tertentu sehingga tidak memerlukan *tunning* parameter dalam melakukan proses otomassi[24].

2.2.8. Algoritma Hill Climbing

Algoritma *hill climbing* merupakan proses pengujian dengan cara pemilihan fungsi *heuristic*[26]. *Hill climbing* adalah teknik optimasi matematis yang termasuk dalam ketegori *local search*[27].

Hill climbing merupakan algoritma iteratif yang dimulai dengan solusi sewenang-wenang untuk sebuah masalah, kemudian mencoba untuk menemukan solusi yang lebih baik dengan mengganti satu elemen solusi secara bertahap[26]. Jika perubahan tersebut menghasilkan solusi yang lebih baik, perubahan tersebut dilakukan pada solusi baru, berulang sampai tidak ada perbaikan lebih lanjut yang dapat ditemukan[26]. Metode *hill climbing* merupakan variasi dari *depth-first search*. Dengan metode ini, eksplorasi terhadap keputusan dilakukan dengan cara *depth-first search* dengan mencari *path* yang bertujuan mengoptimalkan *cost* untuk menuju goal/keputusan[27].

Secara matematis algoritma *hill climbing* mencoba memaksimalkan (atau meminimalkan) fungsi target f (x), di mana x adalah vektor dari nilai kontinyu atau diskrit. Pada setiap iterasi, *hill climbing* akan menyesuaikan satu elemen dalam x dan menententukan apakah perubahan tersebut meningkatkan nilai f (x). Perbedaan *hill climbing* dengan metode kemiringan gradien yang menyesuaikan semua nilai pada x pada setiap iterasi sesuai dengan gradien *hill*. Dengan

hill climbing, perubahan apapun yang meningkatkan f (x) dapat diterima, dan proses berlanjut sampai tidak ada perubahan yang dapat ditemukan untuk memperbaiki nilai f (x). Kemudian x dikatakan "optimal secara lokal"[26].

Secara umum algoritma *hill climbing* dapat dilihat pada Gambar 2.3 dibawah ini.

```
Discrete Space Hill Climbing Algorithm
   currentNode = startNode;
   loop do
      L = NEIGHBORS(currentNode);
      nextEval = -INF;
      nextNode = NULL;
      for all x in L
         if (EVAL(x) > nextEval)
              nextNode = x;
              nextEval = EVAL(x);
      if nextEval <= EVAL(currentNode)
         //Return current node since no
better neighbors exist
         return currentNode;
      currentNode = nextNode;
```

Gambar 2.3 Pseudocode hill climbing
(Sumber: Russel, 2003)

2.2.9. Algoritma Tabu Search

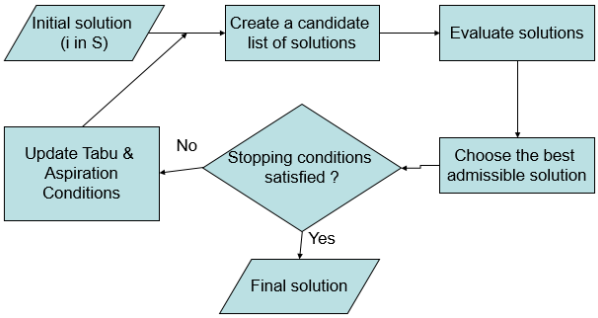
Istilah *Tabu search* berasal dari kata *togan* yaitu suatu bahasa polenesia yang digunakan oleh suku Aboribin pulau Tonga untuk mengidikasikan suatu hal yang tidak diperbolehkan atau diterma karena melanggar kesakralan dan budaya mereka[28]. Kata tabu berarti suatu larangan yang tidak boleh dilakukan karena adanya suatu hal yang berbahaya. Algoritma *tabu search* merupakan sebuah metode optimasi *meta-heuristics* yang berbasis pada *local search* yang pertama kali diperkenalkan oleh Glover pada tahun 1968[3].

Konsep dasar dari algoritma *tabu search* adalah bagaimana mencegah proses pencarian dari *local search* sehingga tidak

akan melakukan pencarian ualng pada solusi yang pernah di telusuri[28]. Algoritma *tabu search* memanfaatkan kombinasi *adaptie memory* dan *expensive exploration*[5]. *Adaptive memory* yang dimliki algoritma *tabu search* dapat memudahkan implementasi prosedur dan pencarian pada *space* tersebut akan lebih efektif dan cepat[5]. Struktur memori yang di dalam *tabu search* dinamakan *tabu list*. *Tabu list* menyimpan atribut dari sebagian transisi solusi yang pernah di temukan dalam iterasi-iterasi sebelumnya. *Tabu list* digunakan untuk menolak semua sousi-solusi yang memenuhi atribut tertentu dalam proses *cyling* pada dareah solusi yang sama[28].

Algoritma *tabu search* memiliki tiga strategi (1)*Forbidding Strategy* yaitu mengontrol apa saja parameter yang akan masuk ke dala tabu list (2)*Freeing strategy* yaitu mengontrol apa yang akan keluar dari *tabu list* dan kapan (3)*Short-term Stategy* yaitu mengatur kolaborasi antara *forbidding strategy* dan *freeing strategy*[29].

Konsep alur kerja algoritma *tabu search* dapat digambarkan seperti Gambar 2.4.



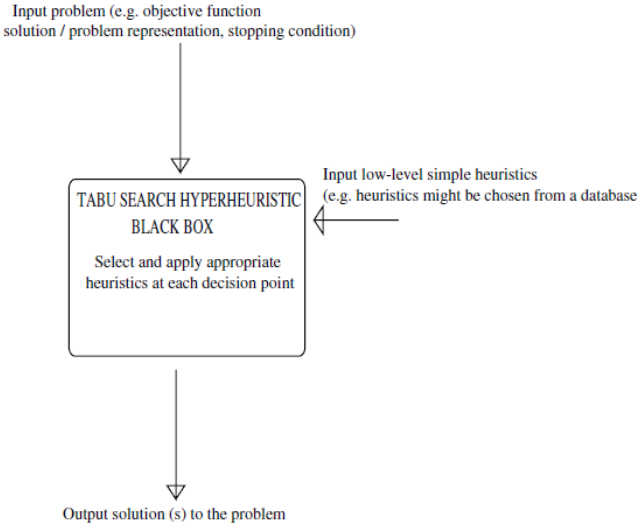
Gambar 2.4 Flow Chart Algoritma Tabu Search
(Sumber: Pham & karaboga, 2000)

Langkah-langkah alur algoritma diatas adalah sebagai berikut: (1) Pilih solusi awal i di S. *Set* i * = i dan k = 0, (2) *Set* k = k +

1 dan buat subset V * dari solusi N (i, k) sehingga salah satu kondisi Tabu dilanggar atau setidaknya satu dari kondisi masih dapat berlaku, (3) Pilih yang terbaik di V * dan *set* i = j, (4) Jika f (i) <f (i *) lalu *set* i * =I, (5) *Update* tabu dan kondisi aspirasi, (6) Jika kondisi sudah terpenuhi maka berhenti proses selesai. Jika tidak maka kembali ke Langkah 2[29].

2.2.10. Tabu Search Hyper-Heuristics

Tabu search hyper heuricstic merupakan penggabungan teknik algoritma *tabu search* dengan *framework hyperheuristics*. Untuk menerapkan *hyper-heuristic* pada masalah tertentu, maka kita hanya membutuhkanya untuk memasukan *low-level heuristic* dan *evaluatiaon function*[5]. *Hyper-heuristic* yang akan digunakan disini bisa dianggap sebagai *black box* yang memberikan solusi terhadap masukan permasalahan[5].



Gambar 2.5 Tabu Search Hyper-Heuristics Black-Box
(Sumber: Burke, dkk, 2004)

Berikut ilustrasi *tabu search hyper heuristic* pada Gambar 2.5. *Black box Hyper heuristic* akan menerima sebagian masukan masalah, infromasi, sebagai low *level heuristic*. Kemudian ditelakukan pemilihan *heurstics* baru dengan menggunakan algoritma *tabu serach,* jika pada suatu kondisi telah terpenuhi kondisi *heuristic* akan keluar sebagai suatu solusi yang paling optimal dari permasalahan[5] sedangka *heuristic* yang tidak terpilih akan dimasukan tabu list sehinga tidak dapat dipilih di dalam langkah iterasi selanjutnya[5].

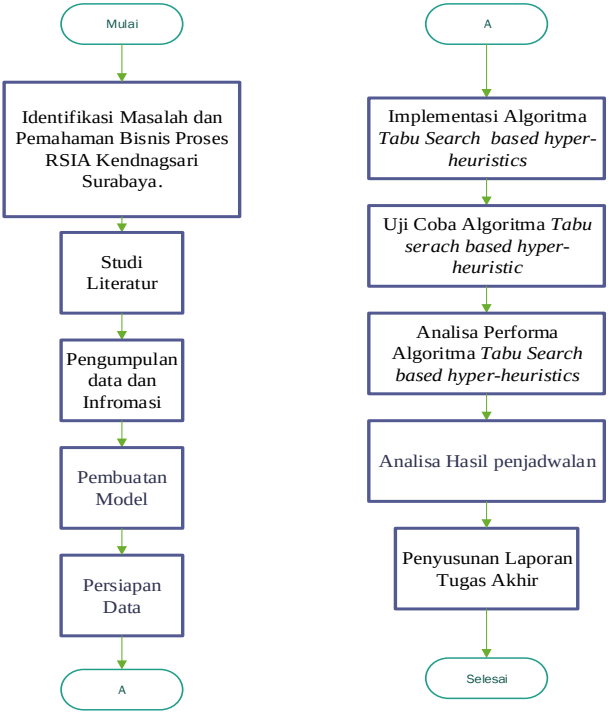
Simple low level heuristics (*e.g. 'add', 'drop', 'swap' objects*) dapat dengan mudah dimasukan ke dalam *black box* sebagai *input*[5]. Akan tetapi terdapat kekurangan pada argumen ini adalah tidak dapat digunakan untuk pengambilan keputusan yang lebih luas karena membutuhkan masukan yang lebih spesifik yang membutuhkan penegtahuan khusus. Akan tetapi argumen ini dapat dengan muda dipatahkan dengan cara mengembangkan sistem yang secara langsung menghubungkan *tabu list* yang ada di *low level* dengan *hyper-heuristics*[5]. Dengan itu masing-masing *heuristic* pada *databased* dijelaskan dengan bahasa general yang sederhana sehingaa pengguna dapat mengerti apa yang mereka lakukan[5].

BAB III
METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengena alur metode penelitian yang akan dilakukan oleh peneliti dalam melakukan penelitian ini. Metode penelitian ini juga digunakan sebagai pedoman untuk melaksanakan penelitian agar terarah dan sistematis.

3.1. Metodologi Penelitian

Diagram Metodologi dari Tugas Akhir ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Metodologi Penelitian

3.2. Tahapan Pelaksanaan Tugas Akhir

Tahapan pelaksanaan tugas akhir mrnjelaskan tentang hal apa saja atau langkah apa saja yang ditempuh penulis untuk menyelesaikan Penelitian tugas akhir ini.

3.2.1. Identifikasi Masalah dan Pemehaman Proses Bisnis

Tahap pertama yang harus dilakukan dalam pengerjaan tugas akhir ini adalah identifikasi masalah yang akan menjadi topik penelitian dalam tugas akhir. Identifikasi masalah dalam penelitian ini adalah permasalahan penjadwalan staf yang ada di RSIA Kedangsari Surabaya. Identifikasi permasalahan digunakan untuk memahami proses bisnis dan sistem yang berjalan di rumah sakit. Setelah menentukan topik permasalahan yang akan dipecahkan maka perlu adanya kajian untuk memahami proses bisnis atau sistem yang berjalan di RSIA Kendangsari Surabaya**.**

3.2.2. Studi Literatur

Pada tahapan ini peneliti mengumpulkan berbagai sumber informasi yang terkait dengan permasalahan penjadwalan staf di RSIA Kendangsari Surabaya. Studi literatur dilakukan untuk memahami objek dan metode yang akan digunakan pada penelitian ini. Studi literatur dapat didapatkan dari berbagai sumber seperti blog, halaman website RSIA Kendansari, buku, jurnal, ataupun studi sebelumnya mengenai topik penjadwalan perawat dan staf rumah sakit. Hal hal yang harus dipelajari oleh peneliti pada tahapan ini adalah konsep *healthcare staff rostering*, *nurse rostering*, algoritma *tabu search*, algoritma *hyper heuristic*, dan algoritma *tabu search hyper heursitics* sesuai dengan yang tertulis pada sub-bab 2.2.5. Dari studi litaratur ini penulis akan mengetauhi karakteristik metode yang akan digunakan dan faktor faltor yang mempengaruhi dalam kasusu penelitian ini.

3.2.3. Pengumpulan Data dan Infromasi

Setelah masalah didefinisikan dan dianalisah dengan jelas peneliti harus melakukan pengumpulan data dan informasi yang mendukung penyelesaian masalah penelitian. Pengumpulan data dan informasi pada tugas akhir ini dilakukan dengan cara wawancara kepada pihak RSIA Kedungsari Surabaya terkait dengan sistematika penjadwalan staf yang telah berjalan saat ini beserta *variable-variable* atau *constraint* yang terjadi. Selain itu pengumpulan data informasi juga digunakan untuk mengambil data penjadwalan perawat sebelumnya. Data dan infromasi yang akan digalih di rumah sakit seperti 1) Regulasi atau aturan penjadwalan Rumah Sakit Ibu dan Anak Kendangsari Surabaya, 2)data staf yang dimiliki, 3)data ruangan atau tempat kerja staf, 3)klasifikasi staf, 4)kebijakan lembur, dan, 5)*healthcare staff preference*.

3.2.4. Pembuatan Model

Pada tahap ini penulis akan membuat persamaan model matematika *Nurse Rostering* dan *Helthcare Rostering* secara umum berdasarkan data dan informasi yang dikumpulkan sebelumnya. Pembuatan model ini dilakukan bertujuan untuk memudahkan pembuatan algoritma yang akan digunakan peneliti. Pembuatan model matematis dilakukan sesuai dengan data set yang diambil dari Rumah Sakit Ibu dan Anak Kendangsari Surabaya. Data tersebut akan dimodelkan secara sistematis dengan mempertimbangkan variabel variable atau *constraint* yang memperngaruhinya. Berikut merupakan langkah langkah pembuatan model (fungsi tujuan, variable keputusan, batasan) sesuai dengan dasr teori teknik manajemen sains pada sub bab 2.2.2:

3.2.5. Persiapan Data

Tahapan ini merupakan tahap perancangan data berdasarkan dta penjadwalan yang terdapat di rumah sakit. Data akan digunakan sebagai *problem domain low level heuristic* pada percobaan ini sehingga dapat diperoleh hasil yang lebih optimal dan yang paling optimal. Data yang digunakan merupakan data staf

seperti jumlah staf, jumlah shif staf, jumlah jam kerja staf serta *constraint-constraint* lain yang harus dipertimbangkan.

3.2.6. Implementasi Algoritma *Tabu Search Based Hyper-Heuristics*

Setelah Pembuatan model matematis permasalahan *nurse rostering* dan *healthcare rostering* maka akan dilakukan implementasi algoritma sesuai dengan permasalahan. Perancangan algoritma akan dibuat sesuai model matematis yang telah dibuat pada tahapan 3.1.4. Perancangan algoritma akan dibangun dengan bahasa pemrogaman java berbasis *local*.

3.2.7. Uji Coba Algoritma *Tabu Search Based Hyper-Heuristics*

Pada tahapan ini dilakukan pengujian dan evaluasi algoritma *tabu search hyper heuristic* yang telah diarancang sebelumnya. Pengujian dilakukan untuk melakukan verifikasi dan validasi algoritma *tabu search hyper-heuristic* yang dirancang. Apakah algoritma yang dirancang sesuai dengan cara kerja *tabu search* dan *hyper-heuristics* sesuai dengan penjelasan pada dasar teori sub-bab 2.2.4. Jika terdapat kesalahan atau *eror* dari algoritma yang dibangun, maka perlu ditinjau ulang.

3.2.8. Analisis Performa Algoritma *Tabu Search Based Hyper-Heuristics*

Pada tahapan ini dilakukan evalusi terhadap hasil penjadwalan yang didapatkan dari algoritma *tabu search hyper heuristic*. Analisi performa dilakukan untuk mendapatkan nilai paling optimumm dari penjadwalan.

3.2.9. Analisis Hasil Penjadwalan Staf Rumah Sakit

Pada tahapan ini dilakukan perbandingan hasil penjadwalan staf di Rumah sakit Ibu dan Anak Kendangsari Surabaya dengan menggunakan algoritma penjadwlaan staf secara otomatis. Analisa yang dilakukan mencakup apakah hasil penjadwalan

merupakan hasil yang paling optimal dan apakah seluruh *hard constraint* dapat diterima.

3.2.10. Penyusunan Laporan Tugas Akhir

Pada tahap terakhir ini merupakan penyusunan laporan tugas akhir untuk melakukan semua dokumentasi yang telah dilakukan selama proses pengerjaan penelitian ini. Format penyusunan laporan tugas akhir ini menyesuasikan ketantuan dari dpartemen sistem infromasi dan Laboratorium Rekayasa Data dan Intelegensia Bisnis (RDIB).

Adapun sistematika penulisan laporan testing adalah sebagai berikut:

BAB I: Pendahuluan

Dalam bab ini akan dijelaskan gambaran umum mengenai tugas akhir yang diangkat. Hal tersebut meliputi latar belakang masalah, perumusan masalah, batasan tugas akhir, tujuan tugas akhir, dan relevan atau manfaat kegiatan tugas akhir. Selain itu, akan dijelaskan pula sistematika penulisan tugas akhir untuk memudahkan pembaca pada saat membaca buku tugas akhir ini.

BAB II: Tinjauan Pustaka

Dalam Bab ini, akan dijelaskan mengenai penelitian terdahulu dan landasan teori yang digunakan sebagai acuan dalam pengerjaan tugas akhir. Penelitian terdahulu merupakan suatu penelitian yang pernah dilakukan oleh peneliti-peneliti sebelumnya yang diguanakan sebagai acuan tugas akhir. Landasan teori merupakan teori-teori yang berhubungan dengan pengerjaan tugas akhir

BAB III: Metodologi Penelitian

Dalam bab ini, akan dijelaskan metodologi yang akan digunakan sebagai panduan untuk menyelesaikan tugas akhir ini.

BAB IV: Perancangan

Dalam bab ini dijelaskan persiapan perancangan aplikasi yaitu dataset dan model berdasarkan batasan atau constraint yang sesuai dengan keadaan komdisi saat ini pada Rumah Sakit Ibu dan Anak kendengsari Surabaya.

BAB V: Implementasi
Bab ini menjelaskan proses pelaksanaan penelitian tugas akhir dan proses implementasi algoritma *tabu search based hyperheuristics* dengan menggunakan bahasa pemrogaman java.

BAB VI: Hasil dan Pembahasan
Pada bab ini menjelaskan mengenai proses uji coba dan analisis terhadap hasil yang diperoleh dari proses implementasi Algoritma *Tabu Search Based Hyper-heuristics*.

BAB VIII: Kesimpulan dan Saran
Pada bagian ini akan dijelaskan hasil implementasi dari rancangan model yang telah dibuat sebelumnya. Bagian ini akan menjelaskan perbandingan hasil antara menggunakan algoritma *tabu search hyper-heuristics* dengan jadwal yang sudah digunakan Rumah Sakit Ibu dan Anak Kendangsari saat ini.

3.3. Jadwal Pelaksanaan Tugas Akhir

Jadwal kegiatan ini akan menjelaskan waktu pengerjaan Tugas Akhir sesuai dengan tahapan yang telah dibuat sebelumnya pada sub bab 3.1. Jadwal kegiatan tugas akhir dapat dilihat pada Tabel 3.1

Tabel 3.1 Jadwal Pelaksanaan Tugas Akhir

No	Kegiatan	Septmber		Oktober				November				Desember			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Identifikasi Masalah dan Pemahaman Proses Bisnis														

No	Kegiatan	Septmber		Oktober				November				Desember			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4
	RSIA Kendangsari														
2	Studi Literatur														
3	Pengumpulan data dan Infromasi														
4	Pembuatan Model														
5	Persiapan Data														
6	Implementasi Algoritma *Tabu Search based hyper-heuristics*														
7	Uji Coba Algoritma *Tabu Search based hyper-heuristics*														
8	Analisa performa Algoritma *Tabu Search based hyper-heuristics*														
9	Analisa Hasil Penjadwalan staf Rumah Sakit														
10	Penyusunan Laporan Tugas Akhir														

Halaman ini sengaja dikosongkan

BAB IV
PERANCANGAN

Dalam bab ini dijelaskan persiapan perancangan aplikasi yaitu data set dan model berdasarkan batasan atau *constraint* yang sesuai dengan keadaan saat ini pada Rumah Sakit Ibu dan Anak kendengsari Surabaya.

4.1. Hasil Pengumpulan Data

Rumah sakit ibu dan anak memilki total 124 staf termasuk tenaga medis, tenaga non medis dan bagian manajemen atau direksi. Dalam memberikan pelayanan terbaik kepada ibu dan anak, RSIA memiliki empat belas unit atau bangsal, enam diantranya ada unit IGD atau rawat jalan, instalasi farmasi, kamar operasi, kamar bayi & NICU, unit Sistem infromasi pendaftaran & Rekam Medis, dan Instalasi gizi.

Proses pembuatan jadwal kerja staff RSIA selama ini masih dilakukan manual. Pembuatan Jadwal tersebut dilakukan secara per unit yang dilakukan oleh masing-masing Penangung Jawab unit. Penangung jawab unit harus mengumpulkan jadwal kepada bagian Sumber daya manusia dan hukum (SDM) RSIA maksimal tanggal 25 setiap bulan untuk bulan berikutnya untuk diinputkan ke dalam sistem mereka.

Penjadwlan staf RSIA memiliki 3 shift dengan pembagian 7 jam yaitu Pagi: 07.00-14.00, Sore: 14.00-21.00, dan Malam: 21.00-07.00. Untuk mengatasi jam kerja yang sibuk pada jam-jam tertentu RSIA menerapkan shift *middle* yang berlangsung dari jam 10.00-17.00 atau jam 12.00-19.00. hal terbut biasanya disesuaikan dengan jadwal praktek dokter dan tanggal-tanggal cantik yang diprediksi akan banyak ibu yang akan melahirkan.

Secara umum penjadwalan dilakukan dengan menggunakan rumus atau Pola umum M-M-LM-L yaitu Staf yang memperoleh shift malam sebanyak 2 kali berturut turut maka tidak boleh memperoleh shift pagi atau siang dan harus

memperoleh libur dihari berikutnya. Setiap staff dalam satu bulan harus mendapatkan bagian shift malam mengikuti pola penjadwalan M-M-LM-L ditetapkan dari bulan sebelumnya. Setelah ditetapkan pola M-M-LM-L maka dilakukan pengeplotan shift pagi dan sift sore secara acak.

Penjadwalan juga memperhitungkan skil yang dimiliki oleh setiap staf. Khusunya perawat RSIA menggolongkan tipe perawat berdasarkan pangkat pelatihan keterampilan yang telah dikuasasi. Untuk perawat anggota merupakan perawat yang memiliki tingkat pelatihan satu atau PK 1 sedangkan untuk bangian kamar operasi atau kepala perawat dibutuhan perawat yang memiliki tinkat pelatihan ketrampilan minimal PK 2 dan PK 3.

Setiap Staf memiliki maksimal jam kerja 7 jam per hari dalam 6 hari per-minggu dan jam lembur dihitung minimal 1 jam dari jadwal yang ditetapkan. Perhitungan jatah minimal atau maksimal jam kerja dalam satu minggu minimal 42 jam kerja. Jika dalam setahun, perhitungan jam kerja tersebut dihitung dari banyaknya jatah libur dan cuti yang diambil oleh staf. Batas minimal libur untuk setiap bulan yang harus didapatkan masing-masing staf harus sama. Setiap bulannya. Apabila tidak sama, maka staf rumah sakit yang mendapat libur lebih sedikit akan diganti hari liburnya di bulan depan. Hal tersebut dilakukan perulangan sampai menemukan total libur dan cuti dalam satu tahun. Untuk pengajuan cuti, masing masing staf harus dilakukan kepada penengungjawab unit atau bangsal disetiap bulan sebelum jadwal tersebut ditunjukan he SDM.

Permasalahan yang dihadapi rumah sakit dalam menjadwlakan adalah pihak SDM kesulitan untuk mengontrol jalanya penjadwalan dikarenakan banyak dari staf rumah sakit yang berganti shift kerja tanpa ada izin terlebih dahulu. Masalah lain juga adanya jam-jam dimana rumah sakit sepi atau sangat ramai. Ketika rumah sakit keadaannya sangat ramai, masalah tersebut biasanya diselesaikan dengan cara mengganti staf rumah sakit

yang bekerja pada shift tersebut pada unit yang renggang ke unit yang padat pasien.

Hal yang ingin dioptimalkan oleh RS adalah bagaimana cara mengontrol jalannya penjdawalan dengan menggunakan sistem yang langsung terintegrasi langsung dengan fingerprint yang ada sehinga kecurangan atau pelanggaran staf rumah sakit akan saling tukar menukar jam dapat diminimalisasi.

4.2. Aturan penjadwalan Rumah Sakit Ibu dan Anak kendangsari Surabaya

Secara umum kebijakan penjadwlan staf setiap unit telah ditetapkan rumah sakit secara resmi sesuai dengan surat ketenagakerjaan dan kebutuhan rumah sakit. Berikut merupakan kebijakan dan regulasi yang ditetapkan oleh pihak SDM RSIA Kendnagsari:

- Shift utama terdiri dari 3 shift yaitu shift pagi (07.00-14.00), shift sore (14.00-21.00), shift Malam (21.00-07.00), dan shift tambahan yaitu shift *Middle* memiliki dua tipe rentan jam kerja (10.00-17.00) dan (12.00-19.00).
- Setiap staf hanya boleh mengisi satu shift per hari
- Setelah shift malam tidak boleh ada shift pagi dan sore
- Pola penetapan shift malam adalah M-M-LM-L
- Setiap staf yang telah mendapatkan shift malam dua kali berturut turut maka harus mendapatkan shift lepas malam kemudian libur
- Staff dapat mengajukan cuti setelah 7 bulan kerja dengan jatah cuti maksimal 1 kali dalam setiap bulan
- Selalu ada minimal satu staf dalam satu shift per hari
- Staf bekerjan 7 jam perhari selama 6 hari sehinga minimal jam kerja yang dimiliki minimal 42 jam per minggu
- Setiap kepala unit harus mendapakan shift pagi setaip hari kerja kantor dan libur di hari minggu

4.3. Data Set

Penjadwalan staf rumah sakit akan dilakukan pada enam unit yang memiliki pola penjadwalan yang lebih unik dari unit yang lain. Keenam unit tersebut yaitu unit IGD & Rawat Jalan, Unit Farmasi, Unit Gizi & Farmasi, Ruang Operrasi, Unit NICU & Ruang Bayi, dan Unit SIM & RM. Data set jadwal staf rumah sakit dijelaskan masing masing unit sesuai dengan kebutuhan dan pola jadwal yang telah ditetapkan saat ini.

4.3.1. Kodifikasi Data set

Kodifikasi data set digunakan untuk memberikan kode atau id data set sehingga lebih mempermudah dalam implementasi data set ke dalam coding algoritma optimasi penjadwalan *tabu serach hyper-heuristics*.

Variabel yang diterjemahkan menjadi kode diataranya adalah Staf id, Skil Id, dan Shift Id. Berikut penjabaran kodifikasi data set.

- Kodifikasi Staf id menggunakan tipe data integer dengan spesifikasi kode terdapat pada Tabel 4.1

Tabel 4.1 Kodifikasi data set

Nama Unit	Kode
Unit Farmasi	1
Unit Farmasi	2
Unit NICU dan Ruang bayi	3
Unit SIM dan RM	4
Unit Gizi dan Café	5
Unit Ruang Operasi	6

Cara penulisan Staf id menggunakan tiga digit integer dengan digit pertama merupakan kode unit dan dikuti

dengan nomor urut di masing masing unit. Contoh 101 merupakan staf unit farmasi

- Kodifikasi Skil Id menggunakan tipe data Integer dengan Spesifikasi kode terdapat pada Tabel 4.2

Tabel 4.2 Kodifikasi Skil Staf

Skil/ pekerjaan	Kode
Kepala Unit	1
Tenaga Ahli	2
Staff Senior	3
Staff	4
Chef	5
Helper	6
Penyaji	7
Cafe	8
Driver	9

- Kodifikasi Shift Id munggunakan tipe data string dengan spesifikasi kode terdapat pada Tabel 4.3.

Tabel 4.3 Kodifikasi Shift Id

Shift	Kode	No Id	Unit
Pagi	P	1	Seluruh Unit
Pagi 1	P1	2	Unit Gizi dan Cafe
Pagi 2	P2	3	Unit Gizi dan Cafe
Pagi Siang	PS	4	Unit Gizi dan Cafe
Siang	S	5	Seluruh Unit
Middle	MS	6	Seluruh Unit, selain Kamar Operasi
Middle 2	MD	7	Unit Gizi dan café, Unit IGD dan rawat jalan
Malam	M	8	Seluruh Unit, selain Kamar Operasi

Shift	Kode	No Id	Unit
Lepas Malam	LM	9	Seluruh Unit, selain Kamar Operasi
Cuti	CT	10	Seluruh Unit

4.3.2. Unit Farmasi

Unit farmasi memiliki 8 staf yang terdiri dari satu apotker, satu petugas Gudang obat, satu staf ahli, dan lima staf anggota. Berikut data staf unit farmasi pada Tabel 4.4:

Tabel 4.4 Kodifikasi unit Farmasi

NO	Staff ID	Skil/ Pekerjaan	Skil ID
1	101	Apoteker Farmasi/ Kepala Unit	1
2	102	Staff Senior	3
3	103	Staf	4
4	104	Staf	4
5	105	Staf	4
6	106	Staf	4
7	107	Staf	4
8	108	Petugas Gudang Obat/tenaga ahli	2

Untuk memnuhi kebutuhan, unit farmasi memiliki ketentuan tertentu untuk memebuhi kebutuhan tersebut. Berikut aturan khusus yang diterapkan unit Farmasi untuk melakukan pejadwalan staf:

- Apoteker dan petugas gudang harus selalu mendapatkan shift pagi dan libur di hari minggu
- Staf senior hanya boleh mendapatkan shift siang dan *middle* shift
- Staf yang memiliki id 103 sampai 107 memiliki pola penjadwalan malam dua kali, lepas malam kemudian libur atau dikenal dengan pola “M-M-LM-L”.

- Staf yang memiliki id 102 sampai 107 juga memiliki shift MS atau *middle shift* sewaktu-waktu apabila *constraint* perhari telah terpenuhi.

4.3.3. Unit Niccu dan Ruang Bayi

Unit NICU dan Ruang bayi memiliki 12 staf yang terdiri kepala unit dan lima staf anggota. Berikut data staf unit NICU dan Ruang Bayi pada Tabel 4.5:

Tabel 4.5 Kodifikasi unit Nicu dan Ruang Bayi

No	ID Staff	Skil/Pekerjaan	ID Skil
1	201	Kepala Unit	1
2	202	Staf	4
3	203	Staf	4
4	204	Staf	4
5	205	Staf	4
6	207	Staf	4
7	208	Staf	4
8	209	Staf	4
9	210	Staf	4
10	211	Staf	4
11	212	Staf	4
12	213	Staf	4

Untuk memnuhi kebutuhan, unit NICU dan Ruang Bayi memiliki syarat dan ketentuan tertentu. Berikut aturan yang diterapkan unit NICUdan Ruang Bayi untuk melakukan pejadwalan staf:

- Setiap kepala unit hanya memiliki shift pagi saja dan libur pada hari *weekend* yaitu hari minggu.

- Staf yang memiliki id 202 sampai 213 memiliki pola penjadwalan malam dua kali, lepas malam kemudian libur atau dikenal dengan pola “M-M-LM-L”.
- Staf yang memiliki id 202 sampai 213 juga memiliki shift MS atau *middle shift* sewaktu-waktu apabila *constraint* perhari telah terpenuhi.

4.3.4. Unit SIM dan RM

Unit SIM dan RM memiliki enam staf yang seluruhnya memiliki skil kerja yang sama. Berikut data staf unit SIM dan RM pada Tabel 4.6:

Tabel 4.6 Kodifikasi unit SIM dan RM

NO	ID Staff	Skil/ Pekerjaan	ID Skil
1	301	Staf	4
2	302	Staf	4
3	303	Staf	4
4	304	Staf	4
5	305	Staf	4
6	306	Staf	4

Untuk memnuhi kebutuhan, unit SIM dan RM tidak memiliki syarat dan ketentuan tertentu. Hal tersebut dikarenakan skil dan setiap kemampuan staf berada pada level yang sama dan diperlukan selama 24 jam sehingga pembagian shift akan merata secara adil anatar shift pagi, siang, malam, dan lepas malam. Sehingga ketentuan penjadwalan yang diterapkan pada unit SIM dan RM adalah:

- Staf yang memiliki id 301 sampai 306 memiliki pola penjadwalan malam dua kali, lepas malam kemudian libur atau dikenal dengan pola “M-M-LM-L”.

- Staf yang memiliki id 301 sampai 306 juga memiliki shift MS atau MD yaitu *middle shift* sewaktu-waktu apabila *constraint* perhari telah terpenuhi.

4.3.5. Unit IGD dan Rawat Jalan

Unit IGD dan Rawat Jalan memiliki 8 staf yang seluruhnya memiliki skil kerja sama. Berikut data staf unit IGD dan Rawat Jalan pada Tabel 4.7:

Tabel 4.7 Kodifikasi unit IGD dan Rawat Jalan

NO	ID Staff	Skil/Pekerjaan	Id Skil
1	401	Staf	4
2	402	Staf	4
3	403	Staf	4
4	404	Staf	4
5	405	Staf	4
6	406	Staf	4
7	407	Staf	4
8	408	Staf	4

Untuk memnuhi kebutuhan, unit IGD dan Rawat Jalan tidak memiliki syarat dan ketentuan tertentu. Hal tersebut dikarenakan skil dan setiap kemampuan staf berada pada level yang sama dan diperlukan selama 24 jam sehingga pembagian shift akan merata secara adil antara shift pagi, siang, malam, dan lepas malam. Sehingga ketentuan penjadwalan yang diterapkan pada unit IGD dan Rawat Jalan adalah:

- Staf yang memiliki id 401 sampai 408 memiliki pola penjadwalan malam dua kali, lepas malam kemudian libur atau dikenal dengan pola “M-M-LM-L”.
- Staf yang memiliki id 401 sampai 408 juga memiliki shift MS atau MD yaitu *middle shift* sewaktu-waktu apabila *constraint* perhari telah terpenuhi.

4.3.6. Unit Gizi dan Café

Unit Gizi dan Café memiliki Sembilan belas staf yang dibagi menjadi enam skil kategori tugas yaitu ahli gizi yang merupakan kepala unit gizi, chef, helper chef, Penyaji, Café, dan Driver. Berikut data staf unit Gizi dan Cafe pada Tabel 4.8.

Tabel 4.8 Kodifikasi unit Gizi dan Cafe

NO	Id Staff	Pekerjaan	ID Skil
1	501	Ahli gizi/Kepala Unit	1
2	502	Chef	5
3	503	Chef	5
4	504	Chef	5
5	505	Chef	5
6	506	Helper	6
7	507	Helper	6
8	508	Helper	6
9	509	Helper	6
10	510	Penyaji	7
11	511	Penyaji	7
12	512	Penyaji	7
13	513	Penyaji	7
14	514	Penyaji	7
15	515	Penyaji	7
16	516	Café	8
17	517	Café	8
18	518	Driver	9
19	519	Driver	9

Untuk memnuhi kebutuhan, unit Gizi memiliki ketentuan tertentu untuk memebuhi kebutuhan tersebut. Berikut aturan

khusus yang diterapkan instalasi Gizi untuk melakukan pejadwalan staf:

- Instalasi Gizi memiliki 3 tipe shift yaitu pagi, siang, malam yang dipecah menjadi 6 tiper yaitu
  - P1 = Pagi Satu (04.00-11.30)
  - P2 = Pagi Dua (05.00-12.00)
  - P = Pagi (06.00-13.00)
  - PS = Pagi + Siang (06.00-20.00)
  - M = Malam (21.00-07.00)
  - S = Siang (13.00-20.00)
- Pembagian Shift malam hanya untuk penyaji yang bertugas untuk menghidangkan makanan kepada pasien dan dokter yang sedang bertugas
- Pembagian Shift P1 dan P2 hanya untuk *Chef* dan *helper*
- Pembagian Shift PS merupakan shift yang diperuntukan sopir yang bertugas mengakomodasikan unit gizi & Café bukan ambulan
- Pembagian shift P1 hanya untuk diperumtukan staf laki-laki diakrenakan jam mulai yang terlalu pagi
- Pembagian shift pagi diperuntukan 1 *chef* dan 1 *helper*.
- Pembagian *Middle shift* hanya diperuntukan kepada *Chef*

4.3.7. Unit Ruang Operasi

Unit Ruang Operasi memiliki 6 staf yang seluruhnya memiliki skil kerja yang sama yaitu pendamping dokter untuk melakukan operasi bedah. Berikut data staf unit Ruang Operasi pada Tabel 4.9:

Tabel 4.9 Kodifikasi uit Ruang Operasi

NO	Id Staff	Pekerjaan	ID Skil
1	601	Staf	4
2	602	Staf	4

NO	Id Staff	Pekerjaan	ID Skil
3	603	Staf	4
4	604	Staf	4
5	605	Staf	4
6	606	Staf	4

Untuk memnuhi kebutuhan, unit Ruang Operasi hanya memiliki dua tipe shift yaitu pagi dan siang. Akan tetapi terdapat jadwal yang bersifat *by request* atau *on call* dimana setiap wajib siaga jika terdapat operasi mendadak. Untuk syarat dan ketentuan skil dan setiap kemampuan staf berada pada level yang sama dan diperlukan selama 24 jam sehingga pembagian shift akan merata secara adil. Sehingga ketentuan penjadwalan yang diterapkan pada unit Ruang Operrasi adalah setiap staf memiliki tipe shift siang dan pagi.

4.4. Proses Pembuatan Model

Permasalahan penjadwlan rumah sakit yang akan dipecahkan adalah bagaimana menjadwalkan secara optimal masing-masing staf di setiap unit selama satu bulan dengan komposisi hari libur dihitung dengan banyaknya hari minggu pada bulan desember 2017 dan masing-masing *hard constraint* yang sudah ditetapkan masing masing unit.

4.4.1. Fungsi Tujuan

Tujuan dari pemodelan permasalahan ini adalah untuk memaksimalkan nilai kepuasan staf dengan meningkatkan keadilan dalam pembagian hari libur antar staf.

4.4.2. Batasan

Penyelesaian permasalahan penjadwalan perawat dengan menggunakan algoritma *tabu search hyperheuristics* mengkategorisasikan setiap *hard constraint* pada masing-masing unit sehingga dapat memenuhi kebutuhan dan

spesifikasi masing-masing unit. Batasan yang ditentukan ada dua yaitu *hard constraint* dan *soft constraint*.

Soft Contraint merupakan ukuran *Optimality* hasil penjadwalan. Ukuran *optimality* yang digunakan disini merupakan nilai *jain fairness index* pada masing masing unit. Sehingga *soft contraints* pada penelitiahan optimasi penjadawalan ini adalah nilai *jain Fairness index*.

Nilai *jains fairness index* dihitung dari skor hari libur masing masing staf dalam satu unit. Shift libur dikategorikan menjadi tiga yaitu libur di hari minggu, libur di hari sabtu dan libur di hari kerja. Kategori libur akan diberikan pembobota dimana libur di hari minggu mendapatakna skor sebesar 4, libur di hari sabtu mendapatkan skor sebesar 2 dan libur pada hari kerja mendapatkan skor sebesar 1.

Batasan-batasan (*Hard Constraint*) yang berlaku pada setiap unit dapat dilihat pada Tabel 4.10:

Tabel 4.10 Daftar *Hard Cosntraint*

No	Unit	Hard Constraint	Kode Hard Constraint
1	Farmasi	Jumlah shift pagi setiap hari harus sama dengan tiga	A1
		Jumlah shift siang setiap hari harus sama dengan dua	A2
		Jumlah shift malam setiap hari harus sama dengan satu	A3
		Jumlah shift pagi pada hari minggu harus sama dengan satu	A4
2	NICU Ruang Bayi	Jumlah shift pagi setiap hari harus lebih dari samadengan dua dan kurang dari samadengan empat	B1

No	Unit	Hard Constraint	Kode Hard Constraint
		Jumlah shift siang setiap hari harus lebih dari samadengan satu dan kurang dari samadengan dua	B2
		Jumlah shift malam setiap hari harus lebih dari sama dengan dua dan kurang dari samadengan tiga	B3
3	SIM & RM	Jumlah shift pagi setiap hari lebih dari sama dengan satu atau kurang dari sama dengan dua	C1
		Jumlah shift siang setiap hari lebih dari sama dengan satu atau kurang dari sama dengan dua	C2
		Jumlah shift malam setiap hari harus sama dengan satu	C3
4	IGD dan POLI	Jumlah shift pagi setiap hari lebih dari sama dengan satu atau kurang dari sama dengan dua	D1
		Jumlah shift siang setiap hari harus lebih dari samadengan dua dan kurang dari samadengan empat	D2
		Jumlah shift malam setiap hari harus berjumlah satu atau dua	D3
5	Gizi	Jumlah shift Ps untuk pada driver harus berjumlah dua pada hari Minggu	E1

No	Unit	Hard Constraint	Kode Hard Constraint
		Setiap hari harus memiliki satu shift malam untuk staf yang berskil sebegai penyaji	E2
6	OK	Jumlah shift pagi setiap hari harus lebih dari samadengan dua dan kurang dari sama dengan empat	F1
		Jumlah shift siang setiap hari harus kurang dari samadengan dua atau kurang dari samadengan tiga	F2
		Jumlah shift pagi pada hari Minggu harus terdiri dari empat	F3

4.4.3. Asumsi Notasi

Penjadwalan perawat dilakukan dengan mengkategorisasikan tipe skil perawat setiap unit. Jadwal diasumsikan selama satu bulan yang memiliki pajang 30, 31, 28 atau 29 hari. Berikut notas notasi yang digunakan dalam penjadwlan otomatis:

i = Staf yang dijadwalkan

d = Hari yang dijadwalkan

t = Tipe shift yang

s = Jumlah hari minggu dalam periode penjadwalan

4.4.4. Variabel Keputusan

Berikut Variabel Keputusan dari Penelitian ini:

$\mathbf{X_{idt}}$ = bernilai 1 jika staf i ditugaskan pada shift **t** pada hari **d**, bernilai 0 jika tidak

Dimana:

$i = 1,2,3, \ldots . n$ n adalah jumlah staf

$d = 1,2,3, \ldots ..$ k; k adalah jumlah hari periode penjadwalan

$t = 1,2,3, \ldots \ldots . . z$; z adalah tipe shift (lihat Tabel 4.3).

4.4.5. Perumusan Batasan

Batasan yang dirumuskan berdasarkan *hard constraint* setiap unit sebegai berikut:

4.4.5.1. Batasam Unit Farmasi

Berikut adalah *hard Constraint* untuk penjadwalan pada unit Farmasi:

Hard Constraint 1

Jumlah staf yang bekerja pada shift pagi setiap hari harus sama dengan tiga. Dirumuskan: untuk setiap nilai d,

$$\sum_{n=1}^{n} \boldsymbol{X_{id1}} = \mathbf{3} \qquad (4.1)$$

Hard Constraint 2

Jumlah staf yang bekerja pada shift siang setiap hari harus sama dengan dua. Dirumuskan: untuk setiap nilai d,

$$\sum_{n=1}^{n} \boldsymbol{X_{id5}} = \mathbf{2} \qquad (4.2)$$

Hard Constraint 3

Jumlah staf yang bekerja pada shift malam setiap hari harus sama dengan satu. Dirumuskan: untuk setiap nilai d,

$$\sum_{n=1}^{n} X_{id8} = 1 \qquad (4.3)$$

Hard Constraint 4
Jumlah staf yang bekerja pada shift pagi pada hari minggu harus sama dengan satu, Dirumuskan: untuk setiap nilai d yang bertepatan di hari minggu.

$$\sum_{n=1}^{n} X_{id1} = 1 \qquad (4.4)$$

4.4.5.2. Batasan Nicu dan Ruang Bayi
Berikut adalah *hard Constraint* untuk penjadwalan pada unit Nicu dan Ruang bayi:

Hard Constraint 1
Jumlah staf yang bekerja pada shift pagi setiap hari harus lebih dari samadengan dua dan kurang dari sama dengan empat. Dirumuskan: untuk setiap nilai d.

$$2 \leq \sum_{n=1}^{n} X_{id1} \leq 4 \qquad (4.5)$$

Hard Constraint 2
Jumlah staf yang bekerja pada shift siang setiap hari harus lebih dari samadengan satu dan kurang dari samadengan dua. Dirumuskan: untuk setiap nilai d,

$$1 \leq \sum_{n=1}^{n} X_{id5} \leq 2 \qquad (4.6)$$

Hard Constraint 3
Jumlah staf yang bekerja pada shift malam setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan tiga. Dirumuskan: untuk setiap nilai d,

$$2 \leq \sum_{n=1}^{n} X_{id8} \leq 3 \qquad (4.7)$$

4.4.5.3. Batasan SIM dan RM

Berikut adalah *hard Constraint* untuk penjadwlan pada unit SIM dan RM:

Hard Constraint 1

Jumlah staf yang bekerja pada shift pagi setiap hari lebih dari sama dengan satu atau kurang dari sama dengan dua. Dirumuskan: untuk setiap nilai d.

$$\mathbf{1} \leq \sum_{n=1}^{n} \boldsymbol{X_{id1}} \leq \mathbf{2} \qquad (4.8)$$

Hard Constraint 2

Jumlah staf yang bekerja pada shift siang setiap hari lebih dari sama dengan satu atau kurang dari sama dengan dua. Dirumuskan: untuk setiap nilai d,

$$\mathbf{1} \leq \sum_{n=1}^{n} \boldsymbol{X_{id5}} \leq \mathbf{2} \qquad (4.9)$$

Hard Constraint 3

Jumlah staf yang bekerja pada shift malam setiap hari harus sama dengan satu. Dirumuskan: untuk setiap nilai d,

$$\sum_{n=1}^{n} \boldsymbol{X_{id8}} = \mathbf{1} \qquad (4.10)$$

4.4.5.4. Batasan IGD dan Rawat Jalan

Berikut adalah *hard Constraint* untuk penjadwalan pada unit IGD dan Rawat Jalan:

Hard Constraint 1

Jumlah staf yang bekerja pada shift pagi setiap hari harus sama dengan satu atau kurang dari sama dengan dua. Dirumuskan: untuk setiap nilai d.

$$\mathbf{1} \leq \sum_{n=1}^{n} \boldsymbol{X_{id1}} \leq \mathbf{2} \qquad (4.11)$$

Hard Constraint 2
Jumlah staf yang bekerja pada shift siang setiap hari harus lebih dari samadengan dua dan kurang dari samadengan empat. Dirumuskan: untuk setiap nilai d,

$$\mathbf{2} \leq \sum_{n=1}^{n} \boldsymbol{X_{id5}} \leq \mathbf{4} \qquad (4.12)$$

Hard Contraint 3
Jumlah staf yang bekerja pada sihft malam setiap hari harus lebih dari sama dengan satu atau kurang dari sama dengan dua. Dirumuskan: untuk setiap nilai d,

$$\mathbf{1} \leq \sum_{n=1}^{n} \boldsymbol{X_{id8}} \leq \mathbf{2} \qquad (4.13)$$

4.4.5.5. Batasan Unit Gizi dan Café
Berikut adalah *hard Constraint* untuk penjadwalan pada unit Gizi dan Cafe:

Hard Constraint 1
Jumlah staf yang bekerja pada shift Ps untuk harus berjumlah dua pada hari Minggu. Dirumuskan: untuk setiap nilai d yang bertepatan di hari minggu.

$$\sum_{n=1}^{n} \boldsymbol{X_{id4}} = \mathbf{2} \qquad (4.14)$$

Hard Constraint 2
Jumlah staf yang bekerja pada shift malam setiap hari harus berjumlah sama dengan satu. Dirumuskan: untuk setiap nilai d,

$$\sum_{n=1}^{n} \boldsymbol{X_{id8}} = \mathbf{1} \qquad (4.15)$$

4.4.5.6. Batasan Ruang Operasi

Berikut adalah *hard Constraint* untuk penjadwalan pada unit Ruang Operasi:

Hard Constraint 1

Jumlah staf yang bekerja pada shift pagi setiap hari harus lebih dari samadengan dua dan kurang dari sama dengan empat. Dirumuskan: untuk setiap nilai d.

$$\mathbf{2} \leq \sum_{n=1}^{n} \boldsymbol{X_{id1}} \leq \mathbf{4} \qquad (4.16)$$

Hard Constraint 2

Jumlah staf yang bekerja pada shift siang setiap hari harus lebih dari sama dengan dua dan kurang dari sama dengan tiga. Dirumuskan: untuk setiap nilai d,

$$\mathbf{2} \leq \sum_{n=1}^{n} \boldsymbol{X_{id5}} \leq \mathbf{3} \qquad (4.17)$$

Hard Constraint 3

Jumlah staf yang bekerja pada shift pagi pada hari Minggu harus terdiri dari empat. Dirumuskan: untuk setiap nilai d yang bertepatan di hari minggu.

$$\sum_{n=1}^{n} \boldsymbol{X_{id1}} = \mathbf{4} \qquad (4.18)$$

Batasan yang dirumuskan berdasarkan *soft constraint* adalah nilai *jain fairness index (JFI).*

$$\boldsymbol{JFI} = \frac{(\sum_{i=1}^{n} \boldsymbol{f_i})^2}{(\boldsymbol{n} * \sum_{i-1}^{n} (\boldsymbol{f_i})^2)} \qquad (4.19)$$

$$\boldsymbol{f_i} = \sum_{d=1}^{k} \sum_{t=1}^{z} |\boldsymbol{X_{idt}} - \mathbf{1}| \, (\mathbf{4}\boldsymbol{M_d} + \mathbf{2}\boldsymbol{S_d} + \boldsymbol{W_d}) \qquad (4.20)$$

$$M_{\mathrm{d}} + S_{\mathrm{d}} + W_{\mathrm{d}} = \mathrm{s} \qquad (4.21)$$

$$M_d \begin{cases} 1, jika\ d\ adalah\ hari\ minggu \\ 0, jika\ tidak \end{cases}$$

$$S_d \begin{cases} 1, jika\ d\ adalah\ hari\ sabtu \\ 0, jika\ tidak \end{cases}$$

$$W_d \begin{cases} 1, jika\ d\ adalah\ hari\ selain\ sabtu\ dan\ minggu \\ 0, jika\ tidak \end{cases}$$

4.4.6. Perumusan Fungsi Tujuan

Fungsi Tujuan yang dirumuskan dalam penjadawalan adalah sebegai berikut:

$$\boldsymbol{MAX}\ JFI = \frac{(\sum_{i=1}^{n} f_i)^2}{(n * \sum_{i-1}^{n} (f_i)^2)}$$

Dimana f_i merupakan persamaan (4.20).

4.5. Pemodelan Algoritma Tabu Search

Pemodelan Algoritma Tabu search yang digunakan disini mengacu pada *pseudocode* pada Penelitian Fred Glover, Manuel Laguna "*Principles of Tabu Search*"[3]. Berikut pseudo code yang digunakan pada Gambar 4.1.

Apply TS short term memory

Apply an elite selection strategy.

do {

Choose one of the elite solutions.

Resume short term memory TS from chosen solution.

Add new solutions to elite list when applicable.

} while (iterations < limit and list not empty)

Gambar 4.1 Pseudocode Tabu search
(Sumber: Glover, 2005)

Sesuai dengan *pseudocode* diatas algoritma *tabu search* dibangun dengan langkah-langkah menetapkan *short memory*, kemudian melakukan pemilihan solusi terbaik dengan cara memilih *elite solution* dan menyimpan *elite sollut*ion pada *short memory*, kemudian dilakukan pemilihan *elite solution* yang baru yang lebih baik dan juga *applicable*. Solusi yang tidak terpilih lagi akan disimpan dalam tabu list atau *short memory*.

BAB V
IMPLEMENTASI

Bab ini menjelaskan proses pelaksanaan penelitian tugas akhir dan proses implementasi algoritma *tabu search based hyperheuristics* dengan menggunakan bahasa pemrogaman java.

5.1. Lingkunngan Uji Coba

Lingkungan Uji coba membahas mengenai lingkungan pengujian dan pengimplemntasian model matematis yang digunakan untuk mengoptimasikan penjadwlan staf rumah sakit pada tugas akhir ini meliputi perangkat keras dan perangkat lunak yang digunakan. Spesifikasi dari implementasi algoritma ditunjukkan pada Tabel 5.1:

Tabel 5.1 Spesifikasi perangkat keras

Perangkat Keras	Spesifikasi
Jenis	Thosiba Satelite CC5-C
Processor	Intel Inside Core i7
RAM	4.00 GB
Hard Disk Drive	1000 GB

Lingkungan perangkat lunak yang digunakan dalam uji coba implementasi metode ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi perangkat lunak

Perangkat Lunak	Fungsi
Windows 10 64 bit	Sistem Operasi
Microsoft Excel 2016	Pengolahan data
Netbeans 8.01	Implementasi algoritma
Microsoft Word 2016	Penulisan Laporan

5.2. Pembuatan *Feasible Schedule*

Pada sub-bab ini menjelaskan pengimplementasian jadwal yang layak berdasarkan pola jadwal pada bulan sebelumnya. Pada sub-bab ini akan dilakukan pembacaan data set setiap unit. Langkah langkah pembuatan *Feasible Schedule* dikerjakan pada halaman kerja *netbeans*.

5.2.1. Membaca Data Set

Data set yang digunakan harus dapat dibaca oleh sistem optimasi yang akan dibangun sehingga diperlukan algoritma sendiri untuk membaca data set. Algoritma yang digunakan untuk membaca data set diimplementasikan dengan menggunakan bahasa pemrograman java.

Berikut merupakan *source code* yang digunakan untuk membaca data set dijelaskan pada Gambar 5.1.

```
1.    int tahun = 2017;
2.          int LengthOfMonth[] = {31,28,31,30,31,30,31,31,30,31,30,31};
3.
4.          int NumberOfEmployee;
5.          int NumberOfDays = 30;
6.          int FirstDay = 3;
7.          int newMonths = 11;
8.          int StartingPoint=0;
9.          String roster[][];
10.         String csvFile;
11.
12.         Random rand = new Random();
13.       public New(int NumberOfEmployee, String csvFile) {
14.         if(tahun%400 == 0 || tahun%4==0){
15.            LengthOfMonth[1] = 29;
16.            }
17.         this.NumberOfEmployee = NumberOfEmployee;
18.         this.csvFile = csvFile;
19.         this.roster= new String[NumberOfEmployee][NumberOfDays];
20.      }
```

Gambar 5.1 Kode program inisiasi array untuk table

Langkah pertama yang dilakukan adalah menentukan inisiasi array yang akan digunakan sebagai tempat penyimpanan data set yang dibaca.

Melakukan inisiasi vaeribel yang bersifat integer untuk menentukan panjang dimensi array. Array yang akan dibuat merupakan array 2 dimensi yang memiliki ukuran rows sebanyak jumlah *employee* dan ukuran kolom sebanyak jumlah hari dalam satu bulan.

```
1.  public void bacafile() {
2.      String cvsSplitBy = ",";
3.
4.      try (BufferedReader br = new BufferedReader(new FileReader(csvFile))) {
5.          int baris = 0;
6.          String line;
7.          while ((line = br.readLine()) != null) {
8.              String[] jadwal = line.split(cvsSplitBy);
9.              this.roster[baris] = jadwal;
10.             baris++;
11.         }
12.     } catch (IOException e) {
13.     }
14.
15.     //untuk bulan selanjutnya
16.     int newSize = roster[0].length + LengthOfMonth[newMonths];
17.     String temporary[][] = new String[NumberOfEmployee][newSize];
18.     for (int i = 0; i < temporary.length; i++) {
19.         System.arraycopy(roster[i], 0, temporary[i], 0, roster[i].length);
20.     }
21.
22.     roster = temporary;
23.     for (String[] roster1 : roster) {
24.         for (int j = 0; j < roster[0].length; j++) {
25.             if (roster1[j] == null) {
26.                 roster1[j] = "-1";
27.             }
28.         }
29.     }
30. }
```

Gambar 5.2 Kode Program membaca file type csv

Langkah selanjutnya yaitu membaca data *type file* csv untuk setiap unit yang akan dijawalkan. Algoritma yang digunakan merupakan pemaca tpe data string dengan menggunakan *Try catch Buffer reader*. Dalam Gambar 5.2 , data set yang berhasil dibaca akan disimpan dalam bentuk array yang bernama roster[][]. .

Langkah yang terakhir merupakan melihat apakah langkah penyimpanan data set ke dalam array berhasil dilakukan. Kode program yang pakai merupakan perulangan dalam perulangan

yang menampilakan hasil jadwal dalam bentuk array dua dimensi. Kode program dapat dilihat pada Gambar 5.3.

```
1.  public void CetakArray() {
2.      for (int i = 1; i <= roster[0].length; i++) {
3.          if (i > 30) {
4.              System.out.printf("%6s,", i - LengthOfMonth[newMonths] + 1);
5.          } else {
6.              System.out.printf("%6s,", i);
7.          }
8.      }
9.      System.out.println(" ");
10.     String Days[] = {"Sn", "Sl", "R", "K", "J", "Sb", "M"};
11.     for (int i = 0; i < roster[0].length; i++) {
12.         System.out.printf("%6s,", Days[(i + (FirstDay - 1)) % 7]);
13.     }
14.     System.out.println(" ");
15.     for (String[] roster1 : roster) {
16.         for (int j = 0; j < roster[0].length; j++) {
17.             if (roster1[j] == null) {
18.                 roster1[j] = "-1";
19.             }
20.             System.out.printf("%6s,", roster1[j]);
21.         }
22.         System.out.println("");
23.     }
24.     System.out.println(" ");
25. }
```

Gambar 5.3 Kode Program menampilkan isi array jadwal

5.2.2. Inisiasi Pseudocode

Pada sub-bab ini akan menjelaskan tentang sifat pola penjadwalan yang dilakukan masing-masing unit. Inisiasi *pseudocode* digunakan untuk memudahkan penerapan algoritma generate secara otomatis jadwal bulan yang akan dioptimasikan dengan jadwal manual bulan sebelumnya.

5.2.2.1. Unit Farmasi

Sifat-sifat atau pola penjadwalan yang dilakukan pada unit farmasi dapat dijelaskan pada langkah langkah pada Gambar 5.4, Gambar 5.5, dan, Gambar 5.6.

1. Kepala Unit dan Petugas Gudang selalu memiliki shift pagi dari hari Senin-Sabtu dan libur setiap hari minggu dan untuk staf id dua memiliki pola shift Siang.

```
for(i=kolom){
    for(baris){
      if (tabel hari minggu)
              tabel_yang_disi [staf][] = Libur
      } else {
              tabel_yang_disi [staf][] = Pagi atau Siang
```

Gambar 5.4 Pseudocode Pola 1unit Farmasi

2. Lihat pola M-M-LM-L yang belum lengkap pada bulan sebelumnya. Isikan pola M-M-LM-L melanjutkan pola bulan sebelumnya untuk Staff berid tiga sampai tujuh.

```
Array [][] Tabel_yang_diisi;

Array [] pola={"M","M","LM","L","P","P","S","S","-1","-1"}


for(kolom){
              for(baris){
              tabel_yang_akan_disi [][] = pola [jarak]
              }

}kolom_mulai++ + jarak Mulai
```

Gambar 5.5 Pseudocode pola 2 unit Farmasi

Setiap hari harus memiliki paling tidak satu orang yang bekerja pada shift malam sehingga jika salah satu staf sudah shift LM maka harus ada staf yang lain shift M. Untuk staf no 3 sampai 7 memiliki pola sehabis shift libur adalah P-P-S-S.

3. Dalam satu hari shift pagi harus terdiri dari 3, shift siang terdiri dari 2. Khusus hari minggu P=1 S=2 M=1 tidak ada middle shift. Jika jumlah Libur, Cuti, dan Lepas malam kurang dari sama dengan satu maka shift sisa merupakan shift middle.

```
for(baris){
    for(kolom){
        for(panjang_pola)
            Check Isi Array
                simpanJumlah array
     }
     //kondisi array
              if (array kosong){
                 if(array[]<1){
                            array = midle shift
```

```
            }else{
                    array = Siang
            }
    }
```

Gambar 5.6 Pseudocode pola 3 unit Farmasi

5.2.2.2. Unit Nicu dan Ruang Bayi

Sifat-sifat atau pola penjadwalan yang dilakukan pada unit Nicu dan Ruang Bayi dapat dijelaskan pada langkah langkah pada Gambar 5.7, dan Gambar 5.8.

1. Kepala unit memiliki shift pagi pada hari senin sampi sabtu dan libur dihari Minggu sehingga isikan nilai “P” pada semua hari kecuali hari Minggu harus disisi dengan nilai “L” yaitu libur.

```
for(i=kolom){
    for(baris){
      if (tabel hari minggu)
            tabel_yang_disi [staf][] = Libur
      } else {
            tabel_yang_disi [staf][] = Pagi atau Siang
```

Gambar 5.7 Pseudocode Pola 1 unit Niccu dan Ruang Bayi

2. Mengsikan Pola M-M-LM-L pada setiap staf unit Niccu dan Ruang bayi kecuali kepala unit. Pengisian pola harus berdasarkan pola pada bulan sebelumnya.
3. Untuk staf dengan id dua sampai staf ber-id tiga belas harus berisikan pola {“P”,“P”,“S”,“S”,“Ms”,“Ms”} setelah mengisi pola M-M-LM-L

```
Array [][] Tabel_yang_diisi;
Array [] pola={"M","M","LM","L","P","P","S","S","-1","-1"}

for(kolom){
      for(baris){
      tabel_yang_akan_disi [][] = pola [jarak]
      }
}kolom_mulai++ + jarak Mulai
```

Gambar 5.8 Pseudocode pola 2 & 3 unit Niccu dan Ruang Bayi

5.2.2.3. Unit SIM dan RM

Sifat-sifat atau pola penjadwalan yang dilakukan pada unit SIM dan RM dapat dijelaskan pada langkah langkah pada Gambar 5.9, dan Gambar 5.10

```
Array [][] Tabel_yang_diisi;
Array [] pola={"M","M","LM","L","P","P","S","S","-1","-1"}

for(kolom){
        for(baris){
        tabel_yang_akan_disi [][] = pola [jarak]
        }
}kolom_mulai++ + jarak Mulai
```

Gambar 5.9 Pseudocode pola 1 & 2 unit Sim dan RM

1. Mengisikan pola M-M-LM-L mengikuti pola sebelumnya pada setiap staf unit SIM dan RM.
   Pengisian pola M-M-LM-Lmengikuti aturan setiap shift malam hanya berisi satu staff. Bila hari tersebut sudah ada orang yang mendapat shift malam, maka staff lain tidak boleh mendapat shift malam.
2. Untuk memnuhi *hard constraint* Pengisian pola M-M-LM-L diikuti dengan pola P-P-P-L-S-S-S di bagian belakangnya.
3. Pengisian pola yang kosong harus didasari dengan ketentuan apabila dalam satu hari jumlah shift pagi, siang, dan malam berjumlah kurang dari sama dengan tiga maka slot jadwal harus diisi dengan “M” shift malam. Apabila berjumlah lebih dari tiga maka diisi dengan nilai P shift pagi.

```
    for(baris){
        for(kolom){
            for(panjang_pola)
                Check Isi Array
               simpanJumlah array
        }

        //kondisi array

           if (array kosong){
```

```
            if(array[]<1){
                        array = malam
            }else{
                        array = pagi
      }
}
```

Gambar 5.10 Pseudocode pola 3 unit Sim dan RM

5.2.2.4. Unit IGD dan Rawat Jalan

Sifat-sifat atau pola penjadwalan yang dilakukan pada unit IGD dan Rawat Jalan dapat dijelaskan pada langkah langkah pada Gambar 5.11.

1. Mengisikan untuk semua staf unit dengan pola M-M-LM-L berdasarkan pola bulan sebelumnya. Setiap minggu setiap staf harus mendapatkan satu kali shift pagi, tiga kali shift sore, satu Libur, dan sisanya merupakan middle shift.

   Total untuk setiap bulan setiap staf harus memiliki empat kali shift pagi, sembilan kali shift siang, empat kali shift libur, empat kali shift malam dan sisnya merupakan *middle* shift. Sehingga pola yang dimasukan merupakan kombinasi shift seperti pola "M-M-LM-L-P-P-S-S-S-S-L-MS-MD-S-S-S".

```
Array [][] Tabel_yang_diisi;
Array [] pola;
for(kolom){
            for(baris){
            tabel_yang_akan_disi [][] = pola [jarak]
            }

}kolom_mulai++ + jarak Mulai
```

Gambar 5.11 Pseudocode pola 1 unit IGD dan Rawat Jalan

5.2.2.5. Unit Gizi dan Cafe

Sifat-sifat atau pola penjadwalan yang dilakukan pada unit Gizi dan Café dapat dijelaskan pada langkah langkah pada Gambar 5.12, Gambar 5.13, dan, Gambar 5.14

1. Ahli Gizi harus memiliki shift pagi dari hari Senin-Sabtu dan libur setiap hari Minggu.

```
for(i=kolom){
    for(baris){
      if (tabel hari minggu)
        tabel_yang_disi [staf][] = Libur
      } else {
       tabel_yang_disi [staf][] = Pagi
```

Gambar 5.12 Pseudocode pola 1 unit Gizi

2. Untuk Staf yang memiliki id dua sampai lima yaitu chef memiliki pola P1-P-S-P1-P-L-S-S-MS-MS.
3. Untuk Staf yang memiliki id enam, delapan, dan Sembilan yaitu *helper chef* memiliki pola P2-P-S-S-L-P-S-P2-P-MS.
4. Untuk staf yang ber-id tujuh memiliki pola S-S-S-P1-P1-P1-L dengan jarak antar pola yaitu tujuh hari. Staff yang ber-id sepuluh memiliki pola S-S-P-P-MD-MD-L dengan jarak antar pola yaitu tujuh hari.

```
Array [][] Tabel_yang_diisi;
Array [] pola={" "}
for(kolom){
      for(baris){
      tabel_yang_akan_disi [][] = pola [jarak]
      }
}kolom_mulai++ + jarak Mulai
```

Gambar 5.13 Pseudocode pola 2, 3, dan 4 unit Gizi

5. Untuk staff 16 dan 19 memiliki pola "PS-S-S-P-P-P-PS-L-P-P-S-S-S-L" dan untuk staff 17 dan 18 memiliki pola "L-P-P-S-S-S-L-PS-S-S-P-P-P-PS"
6. Untuk staf ber id sebelasa sampai lima belas garus memiliki pola M-M-LM-L. langkah pertama yang harus dilakukan adalah meilaht kombinasi pola pada bulan sebelumya yang belum lengkap pada bulan sebelumnya. Setiap hari setidaknya harus memiliki satu orang yang bekerja pada shift malam sehingga jika salah satu staf sudah mengisi shift LM (lepas malam)

maka harus ada staf yang lain yang mengisi shift M. kemudian pola tersebut didkuti dengan pola P-P-S-S.

```
Array [][] Tabel_yang_diisi;
Array [] pola={"M","M","LM","L","P","P","S","S","-1","-1"}
for(kolom){
        for(baris){
        tabel_yang_akan_disi [][] = pola [jarak]
        }
}kolom_mulai++ + jarak Mulai
```

Gambar 5.14 Pseudocode pola 2, 3, 5 & 6 Unit Gizi

7. Untuk staf ber-id sepuluh sampai lima belas, jika masih terdapat slot jadwal yang masih kosong maka dilakukan pengechekan, jika dalam satu hari terdapat shift MD maka slot kosong harus disi dengan nilai P atau shift pagi, dan jika tidak ada shift MD maka slot kosong harus disis shift MD dan jika jumlah Libur, Cuti, dan Lepas malam kurang dari sama dengan satu maka shift sisa merupakan *middle* shift.

```
for(baris){
    for(kolom){
        for(panjang_pola)
            Check Isi Array
          simpanJumlah array
     }


    //kondisi array
        if (array kosong){
            if(array[]<1){
                        array = pagi
            }else{
                        array = MD
        }
    }
```

Gambar 5.15 Pseudocode pola 7 unit Gizi

5.2.2.6. Ruang Operasi

Sifat-sifat atau pola penjadwalan yang dilakukan pada unit Gizi dan Café dapat dijelaskan pada langkah langkah pada Gambar 5.16, Gambar 5.17, dan Gambar 5.18.

1. Mengisikan Pola P-P-S-P-S-L-P-P-S untuk setiap staf unit Ruang Operasi

```
Array [] pola={"P","P","S","P","S","L","P","P","S","-1"}

for(Kolom){
        for(baris){
                tabel_yang_disi [][] = pola [jarak]
        }
}kolom_mulai++
```

Gambar 5.16 Pseudocode Pola 1 unit Ruang Operasi

2. Mengisikan kolom kosong dengan spesifikasi shift siang harus 2, dan shift pagi harus benjumlah kurang dari samadengan tiga.

```
for(baris){
        for(Kolom){
                for(panjang_pola)
                        Check Isi Array
                        simpanJumlah array
                }
                        //kondisi array
                if (array == null){
                        if(array[]<1){
                                array = siang
                        }else{
                                array = pagi
                        }
        }
}
```

Gambar 5.17 Pseuodecode pola 2 unit Ruang Operasi

3. Setiap hari minggu shift pagi harus berjumlah empat dan shift Libur harus berjumlah sebanyak dua. Sehingaa pola yang harus disikan adalah L-L-P-P-P-P secara berulang ke bawah dengan jarak antara staf adalah dua kotak slot jadwal.

```
for(baris){
    for(Kolom){
        tabel_yang_disi []][] = pola [jarak]
                }
    }baris_mulai++
```

Gambar 5.18 Pseudocode pola 3 Ruang Operasi

5.2.3. Implementasi *Feasible Schedule*

Pada sub bab ini akan dijelaskan mengenai pembuatan jadwal otomatis dengan cara membangun kode program dengan bahasa pemogrman java. kode program jadwal otomatis dibangun perunit sesuai dengan pola dan sifat sifat jadwal manual.

5.2.3.1. Unit Farmasi

Kode program pembuatan *feasible schedule* secara otomatis dibangun berdasarkan *pseudocode* yang sesuai dengan pola penjadwalan unit farmasi sehingga jadwal yang dihasilkan dapat memenuhi semua *hard constraint.*

Pola satu memenuhi *hard constraint* bahwa staf dengan id satu yaitu kepala unit dan id delapan merupakan petugas gudang obat-obatan harus selalu mendapatkan shift pagi mulai dari hari senin sampai hari sabtu dan selalu libur di hari minggu. Untuk memenuhi kebutuhan tersebut maka dibangun dua perulangan secara horizontal dan vertical untuk mengisikan shift pagi. sedangkan untuk mengisikan shift siang dilakukan pengechekan jika hari minggu dengan menggunakan perhitungan jarak. jarak antara shift libur dengan libur merupakan tujuh hari sehingga setiap bilangan index array yang habis dibagi tujuh maka akan disikan dengan nilai Libur. Maka kode program yang dibangun dapat dilihat pada Gambar 5.19.

```
1.  for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ;
    i++) {
2.          roster[0][i] = "P";
3.          roster[NumberOfEmployee-1][i] = "P";
4.          if((i+FirstDay)%7 == 0){
5.             roster[0][i]= "L";
6.             roster[NumberOfEmployee-1][i] = "L";
7.          }
8.          }
```

Gambar 5.19 Kode program unit Farmasi pola (1)

Pola satu memenuhi *hard constraint* bahwa staf dengan id dua harus selalu mendapatkan shift siang mulai dari hari senin sampai hari sabtu dan selalu libur di hari minggu. Kode program yang dibangun cukup mirip dengan pola satu. Perbedaanya terletak pada kondisi yang memnuhi untuk nilai shift pagi. kode program yang dibangun dapat dilihat pada Gambar 5.29.

```
1.  for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ;
    i++) {
2.          roster[1][i] = "S";
3.          if((i+FirstDay)%7 == 0){
4.             roster[1][i]= "L";
5.          }
6.         }
```

Gambar 5.20 Kode program unit Farmasi pola (2)

Pola ke tiga merupakan pola yang memenuhi *hard constraint* bahwa setiap staf ber-id tiga sampai tujuh harus memiliki shift pagi, siang, malam, lepas malam, dan libur. Untuk memnuhi *hard constraint* tersebut maka kode program yang dibangun dapat dilihat pada Gambar 5.21.

```
1.  String Pattern []={"M", "M", "LM","L","P","P","S","S","-1", "-1"};
2.      for (int i = 0; i < NumbE; i++) {
3.         for (int j = NumberOfDays; j <
    NumberOfDays+LengthOfMonth[newMonths]; j++) {
4.             roster[i+2][j]= Pattern[(j-StartingPoint)%10];
5.
6.         }
7.         StartingPoint +=2;
8.      }
```

Gambar 5.21 Kode program unit Farmasi pola (3)

Menginisiasikan array pola yang akan disiskan. Memnggunakan perulangan secara vertical dan horizontal untuk mengisi kotal slot jadwal, pengisian didasarkan jarak anatar array pola. Jarak antara pola starting point pertama

dengan perulangan berikunya merupakan sepuluh. Kotak akan berulang meloncati setiap dua kotak secara vertical.

Pola yang kempat merupakan pola yang digunakan untuk memenuhi *hard contrsint* per hari diaman setiap hari unit farmasi harus memiliki komposisi shift pagi sebanyak tiga, shift siang sebanyak 2, shift malam sebanyak 1.

```
1.  String NamaShift[]={"P","S","M","LM","L","CT","MS"};
2.  for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
3.  int JmlShift [] ={0,0,0,0,0,0,0};
4.  for (int i = 0 ; i <NumberOfEmployee; i++) {
5.  for (int k = 0; k < JmlShift.length; k++) {
6.  if(roster[i][j].equalsIgnoreCase(NamaShift[k])) {
7.  JmlShift[k] ++;
8.  }
9.  }
10. if(roster[i][j]== "-1"){
11. if((JmlShift[3]+JmlShift[4]+JmlShift[5])<= 1){
12. roster[i][j]="MS";
13. }else{
14. roster[i][j]="S";
15. }
16. }
17. }
18. }
```

Gambar 5.22 Kode program unit farmasi pola (4)

Maka kode program yang dibangun yaitu melakukan pengechekan jumlah shift yang ada secara horizontal. Perhitungan shift dilakukan dengan cara menyimpan jumlah shift perhari dalam bentuk array. Untuk pengechekan dibuat perulangan secara horizontal dan vertical. Untuk mengisi nilai shift maka harus membuhi kondisi jika jumlah libur, lepas malam, dan cuti kurang dari sama dengan satu. Kode program dapat dilihat pada Gambar 5.22.

5.2.3.2. Unit Nicu dan Ruang Bayi

Kode program pembuatan *feasible schedule* secara otomatis dibangun berdasarkan pseudocode yang sesuai dengan pola penjadwalan unit Nicu dan Ruang Bayi sehingga jadwal yang dihasilkan dapat memenuhi semua *hard constraint*.

Pola pertama yang harus dilakukan adalah mengisikan shift kepala unit. Kepala unit memiliki tipe shift selalu pagi dal libur di hari jum'at. Kode program yang dibangun berbentuk perulangan sebanyak kolom yang akan diisi. Untuk memisahkan hari Minggu dan non hari Minggu digunakan kondisi *if else*, jika kelipatan tujuh maka array dua dimensi roster akan disisi dengan huruf "L", selain itu maka array akan diisi dengan huruf "p". kode program dapat dilihat pada Gambar 5.23.

```
1.  for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ;
    i++) {
2.          roster[0][i] = "P";
3.          if((i+FirstDay)%7 == 0){
4.             roster[0][i]= "L";
5.          }
6.          }
```

Gambar 5.23 Kode program unit Niccu dan Ruang Bayi Pola (1)

Pola kedua yaitu memasukan shift untuk staf id dua sampai staf id dua belas. Pola yang akan dimasukan disimpan dalam array. Untuk melakukan pengisian array dilakukan *double* perulangan sejumlah employee dan sejumlah hari yang akan diisi. Pola yang akan diisikan akan selelu bergeser dua hari antar staf satu dengan staf sesudahnya. Sehingga ditambahkan increment sebanyak 2++. Kode program pola ke dua ini dapat dilihat pada gambar Gambar 5.24.

```
1.  int NumbEmployeeUnitNICU= 12;
2.      String Pattern []={"M", "M", "LM","L","P","P","S","S","MS", "MS"};
3.      for (int i = 0; i < NumbEmployeeUnitNICU; i++) {
4.         for (int j = NumberOfDays; j <
    NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.            roster[i+1][j]= Pattern[(j-StartingPoint)%10];
6.         }
7.         StartingPoint +=2;
8.      }
```

Gambar 5.24 Kode program Unit Niccu dan Ruang Bayi Pola (2)

5.2.3.3. Unit SIM dan RM

Kode program pembuatan *feasible schedule* secara otomatis dibangun berdasarkan *pseudocode* yang sesuai dengan pola penjadwalan unit SIM dan RM sehingga jadwal yang dihasilkan dapat memenuhi semua *hard constraint*.

Pola penjadwalan Unit SIM dan RM harus memiliki pola M-M-LM-L yang seleu berulang. Sedangkan pola P-P-P-L-S-S digunakan untuk memnuhi hard constraint perhari. Sehingga kode program yang dibangun adalah dengan menggunakan perulangan pola "M-M-LM-L-P-P-P-L-S-S". pengisian pola dilakukan berbeda untuk setiap staf yaitu jarak pengisian bergesar setiap dua kotak array dan jarak antara pola pertama dengan sesudahnya merupakan dua belas kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.25.

```
1.  //STEP 1 memasukkan pola M M LM L P P P L S S S untuk semua employee
2.      String Pattern []={"M","M","LM","L","P","P","P","L","S","S","-1","-1"};
3.      for (int i = 0; i < NumberOfEmployee; i++) {
4.          for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.              roster[i][j]= Pattern[(j-StartingPoint)%12];
6.          }
7.          StartingPoint +=2;
8.      }
```

Gambar 5.25 Kode program unit Sim dan RM (1)

Pola kedua digunakan untuk menetapkan middle shift pada satu hari. Fungsi yang dibangun adalah mengechekapakah dalam satu hari sudah memnuhi *hard constraint* atau tidak. Jika telah terdapat shift middle maka array kosong akan berisikan nilai "P" atau pagi. Sedangkan jika jumlah shift pagi ditambah shift malam dan shift siang kurang dari samadengan tiga maka array kosong akan diisi dengan nilai "MS" atau *middle* shift. Kode program pola ini dapat dilihat pada Gambar 5.26.

```
1.  String NamaShift[]={"P","S","M","LM","L","CT","MS"};
2.          for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
3.              int JmlShift [] = {0,0,0,0,0,0,0};
4.              for (int i = 0 ; i < NumberOfEmployee; i++) {
5.                 for (int k = 0; k < JmlShift.length; k++) {
6.                    if(roster[i][j].equalsIgnoreCase(NamaShift[k])) {
7.                       JmlShift[k] ++;
8.                    }
9.                 }
10.                if(roster[i][j]== "-1"){
11.                   if((JmlShift[0]+JmlShift[1]+JmlShift[2]) <= 3){
12.                      roster[i][j]="MS";
13.                   }else {
14.                      roster[i][j]="P";
15.                     // roster[i][j]="S";
16.                   }
17.                }
18.             }
19.         }
```

Gambar 5.26 Kode program unit Sim dan RM pola (2)

5.2.3.4. Unit IGD dan Rawat Jalan

Kode program *generate feasible jadwal* secara otomatis dibangun berdasarkan *pseudocode* yang sesuai dengan pola penjadwalan unit IGD dan Rawat jalan sehingga jadwal yang dihasilkan dapat memenuhi semua *hard constraint*.

Pola yang diterpakan merupakan kombinasi shift “M-M-LM-L-P-P-S-S-S-S-S-L-MD-MS-S-S-S). Pola disiikan setiap staf dengan perbedaan yaitu untuk jarak pengisian bergesar setiap dua kotak array dan jarak antara pola pertama dengan sesudahnya merupakan lima belas kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.27.

```
1.  String Pattern []={"M", "M",
    "LM","L","P","P","S","S","S","S","S","L","MD","MS","S"};
2.      for (int i = 0; i < NumberOfEmployee; i++) {
3.          for (int j = NumberOfDays; j <
    NumberOfDays+LengthOfMonth[newMonths]; j++) {
4.              roster[i][j]= Pattern[(j-StartingPoint)%15];
5.          }
6.          StartingPoint +=2;
7.      }
```

Gambar 5.27 Kode program unit IGD dan Rawat Jalan Pola (1)

5.2.3.5. Unit Gizi

Kode program *generate feasible jadwal* secara otomatis dibangun berdasarkan pseudocode yang sesuai dengan pola penjadwalan unit Gizi dan Café sehingga jadwal yang dihasilkan dapat memenuhi semua *hard constraint*.

Pola pertama yang diterapkan yaitu mengisi shift pagi untuk staf id satu yaitu kepala unit sekaligus ahli gizi. Kode program yang dibangun yaitu menggunakan perulangan dan kondisi *if else* untuk setiap kolom di bulan desember. Jika hari Minggu maka nilai yang dimasukan adalah "L" jika hari kerja biasa maka nilai yang dimasukan adalah "L". Kode Program dapat dilihat pada Gambar 5.28.

```
1.  for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ;
    i++) {
2.          roster[0][i] = "P";
3.          roster[NumberOfEmployee-1][i] = "P";
4.          if((i+FirstDay)%7 == 0){   //minggu
5.             roster[0][i]= "L";     //maka libur
6.             roster[NumberOfEmployee-1][i] = "L";
7.          }
8.         }
```

Gambar 5.28 Kode program unit Gizi pola (1)

Pola ke dua penjadwalan gizi dibangun untuk staf ber id dua samapai lima. Kode program yang dibangun harus memiliki pola "P1-P-S-P1-P-L-S-S-MD-MD". Pengisian pola dilakukan berbeda untuk setiap staf yaitu jarak pengisian bergesar setiap dua kotak array dan jarak antara pola pertama dengan sesudahnya merupakan sepuluh kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.29.

```
1.  int StartingPointChef=0;
2.      int NumbChef= 4;
3.      String PatternChef [] = {"P1","P","S","P1","P","L","S","S","MD","MD"};
4.      for (int i = 0; i < NumbChef; i++) {
5.        for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.          roster[i+1][j]= PatternChef[(j-StartingPointChef)%10];
7.        }
8.        StartingPointChef += 2;
9.      }
```

Gambar 5.29 Kode program unit Gizi pola (2)

Pola ketiga yang diterapkan yaitu mengisi shift untuk staf id enam. Kode program yang dibangun harus memiliki pola "L-P1-P1-S-S-S-P1-P1". Pengisian pola dilakukan menggunakan perulangan dengan jarak pengisisian antara pola pertama dengan sesudahnya merupakan delapan kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.30

```
1.  String Pattern7 [] = {"L","P1","P1","S","S","S","P1","P1"};
2.      for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++) {
3.        roster[5][j]= Pattern7[(j+5)%8];
4.      }
```

Gambar 5.30 Kode program unit Gizi pola (3)

Pola ke empat penjadwalan unit Gizi dibangun untuk staf ber id tujuh samapai sembilan. Kode program yang dibangun harus memiliki pola "P2-P-S-S-L-P-S-P2-P-MS". Pengisian pola dilakukan berbeda untuk setiap staf yaitu jarak pengisian bergesar setiap dua kotak array dan jarak antara pola pertama dengan sesudahnya merupakan sepuluh kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.31

```
1.  int StartingPointHelper=0;
2.      int NumbHelper = 3;
3.      String PatternHelper [] = {"P2","P","S","S","L","P","S","P2","P","MS"};
4.
5.      for (int i = 0; i < NumbHelper; i++) {
6.        for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++) {
7.          roster[i+6][j]= PatternHelper[(j-StartingPointHelper)%10];
8.        }
```

```
9.              StartingPointHelper += 2;
10.         }
```

Gambar 5.31 Kode program unit Gizi pola (4)

Pola ke lima yang diterapkan yaitu mengisi shift untuk staf id sepuluh. Kode program yang dibangun harus memiliki pola "S-S-P-P-MD-MD-L". Pengisian pola dilakukan menggunakan perulangan dengan jarak pengisisian antara pola pertama dengan sesudahnya merupakan tujuh kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.32.

```
1.      String Pattern10 [] = {"S","S","P","P","MD","MD","L"};
2.
3.          for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++) {
4.              roster[9][j]= Pattern10[(j+5)%7];
5.          }
```

Gambar 5.32 Kode program unit Gizi pola (5)

Pola ke enam penjadwalan unit Gizi dibangun untuk staf ber id sebelas sampai lima belas. Kode program harus memiliki pola M-M-LM-L yang seleu berulang. Sedangkan pola P-P-S-S digunakan untuk memnuhi *hard constraint* perhari. Sehingga kode program yang dibangun adalah dengan menggunakan perulangan pola "M-M-LM-L-P-P-S-S". Pengisian pola dilakukan berbeda untuk setiap staf yaitu jarak pengisian bergesar setiap dua kotak array dan jarak antara pola pertama dengan sesudahnya merupakan sepuluh kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.33.

```
1.
2.
3.      int StartingPointPenyaji=0;
4.          int NumbE= 5;
5.          String Pattern [] = {"M","M","LM","L","P","P","S","S","-1","-1"};
6.
7.          for (int i = 0; i < NumbE; i++) {
8.              for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++) {
9.                  roster[i+10][j]= Pattern[(j-StartingPointPenyaji)%10];
10.             }
11.             StartingPointPenyaji += 2;
```

```
12.         }
```

Gambar 5.33 Kode program unit Gizi pola (6)

Pola ke tujuh yang diterapkan yaitu mengisi shift untuk staf id enam belas dan sembilan belas. Kode program yang dibangun harus memiliki pola "PS-S-S-P-P-PS-L-P-P-S-S-S-L". Pengisian pola dilakukan menggunakan perulangan dengan jarak pengisisian antara pola pertama dengan sesudahnya merupakan empat belas kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.34.

```
1.  for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ;
    i++) {
2.          String Pattern1619 [] =
    {"PS","S","S","P","P","P","PS","L","P","P","S","S","S","L"};
3.          for (int j = NumberOfDays; j <
    NumberOfDays+LengthOfMonth[newMonths]; j++) {
4.            roster[15][j]= Pattern1619[(j+9)%14];
5.            roster[18][j]= Pattern1619[(j+9)%14];
6.          }
7.        }
```

Gambar 5.34 Kode program unit Gizi pola (7)

Pola ke tujuh yang diterapkan yaitu mengisi shift untuk staf id tujuh belas dan delapan belas. Kode program yang dibangun harus memiliki pola "L-P-P-S-S-S-L-PS-S-S-P-P-P-PS". Pengisian pola dilakukan menggunakan perulangan dengan jarak pengisisian antara pola pertama dengan sesudahnya merupakan empat belas kotak array. Maka kode program yang dibangun dapat dilihat pada Gambar 5.35.

```
1.  for (int i = NumberOfDays; i <NumberOfDays+LengthOfMonth[newMonths] ;
    i++) {
2.
3.          String Pattern1718 [] =
    {"L","P","P","S","S","S","L","PS","S","S","P","P","P","PS"};
4.
5.          for (int j = NumberOfDays; j <
    NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.            roster[16][j]= Pattern1718[(j+9)%14];
7.            roster[17][j]= Pattern1718[(j+9)%14];
8.          }
9.        }
```

```
10.
```

Gambar 5.35 Kode program unit Gizi pola (8)

Pola penjadawalan ke sembilan pada unit gizi digunakan untuk menentukan shift midlle pada slot kotak array yang masih kosong. Kode program ini akan diaplikasikan untuk staf ber-id sebelas sampai lima belas. Maka kode program yang dibangun yaitu melakukan pengechekan jumlah shift yang ada secara horizontal. Perhitungan shift dilakukan dengan cara menyimpan jumlah shift perhari dalam bentuk array. Untuk pengechekan dibuat perulangan secara horizontal dan vertical. Untuk mengisi nilai shift maka harus membuhi kondisi jika jumlah libur, lepas malam, dan cuti kurang dari sama dengan satu. Kode program dapat dilihat pada Gambar 5.36.

```
1.  String NamaShift[]={"P","S","M","LM","L","CT","MD"};
2.  for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
3.  int JmlShift [] ={0,0,0,0,0,0,0};
4.  for (int i = 0 ; i <NumberOfEmployee; i++) {
5.  for (int k = 0; k < JmlShift.length; k++) {
6.  if(roster[i][j].equalsIgnoreCase(NamaShift[k])) {
7.  JmlShift[k] ++;
8.  }
9.  }
10. if(roster[i][j] == "-1"){
11. if(JmlShift[6] == 0){
12. roster[i][j]="MD";
13. }else{
14. roster[i][j]="P";
15. }
16. }
17. }
18. }
```

Gambar 5.36 Kode program unit Gizi pola (9)

5.2.3.6. Ruang Operasi

Kode program *generate feasible jadwal* secara otomatis dibangun berdasarkan pseudocode yang sesuai dengan pola penjadwalan unit Ruang Operasi sehingga jadwal yang dihasilkan dapat memenuhi semua *hard constraint*.

Pola pertama dilakukan pengisian kotak slot jadwal secara horizontal dengan pola "P-P-S-P-S-L-P-P-S". untuk mengisiskan slot maka dibangun perulangan secara vertical dan horizontal. Pengisian pola dilakukan dengan menggunakan jarak antara starting point pertama dengan starting point kedua yaitu sepuluh dan turun setaip baris satu kotak. Kode program dapat dilihat pada Gambar 5.37.

```
1.  String Pattern []={"P","P","S","P","S","L","P","P","S","-1"};
2.
3.      for (int i = 0; i < NumberOfEmployee; i++) {
4.        for (int j = NumberOfDays; j < NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.          roster[i][j]= Pattern[(j-StartingPoint)%10];
6.        }
7.        StartingPoint ++;
8.  }
```

Gambar 5.37 Kode Program Unit Ruang Operasi pola (1)

Pola ke dua digunakan untuk memenuhi *hard constraint* bahwa setiap hari setidaknya Ruang Operasi harus memiliki dua orang harus berada pada shift siang. Sehingga kode program yang dibangun adalah dengan cara mengechek jumlah masing-masing shift yang telah teralokasikan. Jumlah shift tersebut disimpan dalam bentuk array yaitu JumlahShift []. Kode program dapat dilihat pada Gambar 5.38.

```
1.  String NamaShift[]={"S","P","L","CT"};
2.
3.      for (int j = NumberOfDays; j <NumberOfDays+LengthOfMonth[newMonths]; j++) {
4.
5.          int JmlShift [] ={0,0,0,0};
6.          for (int i = 0 ; i <NumberOfEmployee; i++) {
7.            for (int k = 0; k < JmlShift.length; k++) {
8.              if(roster[i][j].equalsIgnoreCase(NamaShift[k])) {
9.                JmlShift[k] ++;
10.             }
11.           }
12.           if(roster[i][j]== "-1"){
13.             if(JmlShift[0]<1){
14.               roster[i][j]= "S";
15.             }else{
16.               roster[i][j]="P";
17.             }
```

```
18.                 }
19.             }
20.     }
```

Gambar 5.38 Kode Program unit Ruang Operasi pola (2)

Pola ke tiga yang harus dipenuhi yaitu jumlah shift P harus berjumlah empat orang dan shift libur sebanyak dua orang. Sehingga kode program yang dibangun yaitu dengan mengisikan pola L-L-P-P-P-P secara vertical. Kode program dapat dilihat pada Gambar 5.39.

```
1.    int StartingPointLibur=0;
2.          String pola2 []={"L","L","P","P","P","P"};
3.          for (int j = NumberOfDays; j <
      NumberOfDays+LengthOfMonth[newMonths]; j++) {
4.            for (int k =0; k <pola2.length; k++) {
5.              if((j+FirstDay)%7 == 0 ){
6.                roster[k][j]=pola2[(k+StartingPointLibur+2)%6];
7.              }
8.            }StartingPointLibur +=2;
9.          }
```

Gambar 5.39 Kode program unit Ruang Operasi pola (3)

5.2.4. Menghitung Nilai Jain Fairness Index

Pada sub-bab ini dilakukan pembangunan kode program untuk mengukur performa hasil jadwal yang dibuat secara otomatis. Ukuran optimal pada optimasi penjadwalan ini merupakan *jain fairness index* yang digunakan untuk mengukur keadilan setiap staf dalam mendapatkan shift libur.

Untuk menemukan nilai *jain fairness index* masing-masing unit, maka kode kode program yang dibangun pertama kali merupakan cara menghitung nilai libur pada jadwal. Perhitungan kategori libur dibedakan menjadi tiga tipe yaitu tipe hari Minggu yang diberikan pembobotan sebanyak seratus, tipe hari Minggu yang diberikan bobot sebesar limapuluh dan libur pada hari kerja diberikan boot sebesar dua puluh lima.

Kode pogram dibangun dalam bentuk *double* perualangan untuk mencari nilai pada array yaitu perulangan sebanyak baris

dan sebanyak kolom. Untuk menenetukan nilai libur masing masing staf digunakan syntax kondisi *if else* seperti pada Gambar 5.40.

```
1.  for (int i = 0 ; i <NumberOfEmployee; i++) {
2.          int a=0;
3.          int JmlLibur[]= {0,0,0};
4.          int Bobot[]={100,50,25};
5.          int liburXbobot= 0;
6.          int Jml_liburXbobot=0;
7.          double JFIperStaf_Kuadrat= 0;
8.
9.          for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
10.           try{
11.            if(roster[i][j].equalsIgnoreCase("L")){
12.              if((j+FirstDay)%7 == 0){
13.                 JmlLibur[0]++;
14.              }else if ((j+FirstDay)%7 == 6){
15.                 JmlLibur[1]++;
16.              }else{
17.                 JmlLibur[2]++;
18.              }
19.            }
20.           } catch (Exception e) {
21.          }
22.        }
```

Gambar 5.40 Kode program untuk menghitung jumlah shift libur

Perhitungan nilai *jain fairness index* dilakukan untuk mencari nilai keadilan antar staf. Yaitu dengan menggunakan persamaaan matematis *jain fairness index*. Persamaan matematika *jains fainerss index* dapat diterjemahkan dalam kode program pada Gambar 5.41 kode program yang dibangun pertama kali yaitu melakukan pembobotan masing masing jumlah hari libur. Jika hari libur berada pada hari Minggu maka diberikan pembobotan sebanyak empat, jika hari libur pada hari sabtu maka pembobotan yang diberikan sebesar dua, dan jika hari libur terletak pada hari kerja maka pembobotan yang diberikan adalah sebesar satu. Hasil perkaliahan jumlah hari libur dengan pembobot akan masing masing dikuadratkan, kemudian dijumlah. Dan penjulamlahan setiap elemen yang telah dikudratkan.

```
1.    for (int k = 0; k < JmlLibur.length; k++) {
2.                int bb= JmlLibur[k];
3.                liburXbobot =JmlLibur[k]*Bobot[k];
4.                Jml_liburXbobot +=liburXbobot;
5.             }
6.                JFIperStaf_Kuadrat =  Math.pow(Jml_liburXbobot, 2);
7.                JmlhJfiStafkudrat +=JFIperStaf_Kuadrat;
8.                JmlSelStaf +=Jml_liburXbobot;
9.          }
10.          JmlSelStaf_kuadrat = Math.pow(JmlSelStaf, 2);
11.          jfi=(JmlSelStaf_kuadrat
      /(NumberOfEmployee*JmlhJfiStafkudrat));
```

Gambar 5.41 Kode Program perhitungan nilai *jains fairness index*

5.2.5. Pemodelan *Hard Constraint* pada Kode Program

Permodelan *hard constraint* ke dalam kode program bahasa java dilakukan agar dapat memberikan batasan pada saat menginisiasikan atau menghasilkan jadwal secara otomatis berdasarkan bulan sebelumnya. Pemodelan batasan pada kode program dilakukan untuk setiap unit sesuai dengan model matematis batasan yang diinisiasikan pada bab 4.4.5.

Untuk menentukan kebenaran *hard constraint* pada jadwal yang dihasilkan secara otomatis maka perlu dilakukan pengechekan. Untuk memudahkan melakukan pengechekan nilai dari array yang bersisi jadwal hasil *generating* otomatis, maka kode program yang dinisiasikan adalah membentuk *object* atau method pada bahasa pemrograman java dengan tipe nilai balikan merupakan *boolean*. Nilai balikan *boolean* akan menghasilkan nilai satu atau nol.

5.2.5.1. Unit Farmasi

Berikut kode program yang dibangun untuk memberikan batasan pada saat *generating* secara otomatis dan proses optimasi pada unit Farmasi.

Kode program yang dibangun pada Gambar 5.42 adalah membentuk *method* atau fungsi yang bernama cekConstraint dengan nilai tipe balikan *boolean*. Melakukan deklarasi nilai *boolean* pertama kali merupakan false. Untuk melakukan

pengechekan isi array jadwal maka dibangun perulangan sebanyak kolom dan perulangan sebanyak baris.

```
1.  public boolean cekConstraint(String [][] roster){
2.      boolean value1= false;
3.      boolean value2= false;
4.
5.      for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.          String [] Shift ={"P","S","M"};
7.          int [] Jml_Shift = {0,0,0};
8.          for (int i = 0; i < NumberOfEmployee; i++) {
```

Gambar 5.42 Inisiasi fungs boolean unit Farmasi

Kode program pada Gambar 5.43 melakukan perintah untuk melihat komposisi shift per hari. Pengechekan hard constraint dibagi menjadi dua yaitu pada hari minggu dan ari kerja sehingaa kode program yang digunakan adalah *if else*. Sedangkan untuk menemukan jumlah masing-masing shift yang tercakup *hard constraint* digunakan fungsi *equalsignorecase* untuk mengambil nilai pada jadwal.

```
1.  if((j+FirstDay)%7 == 0){
2.              if(roster[i][j].equalsIgnoreCase("P")){
3.                  Jml_Shift[0]++;
4.                } else if(roster[i][j].equalsIgnoreCase("S")){
5.                  Jml_Shift[1]++;
6.                } else if(roster[i][j].equalsIgnoreCase("M")){
7.                  Jml_Shift[2]++;
8.                }
9.
10.             if ((Jml_Shift[0] == 1 && Jml_Shift[1]== 2 )&& Jml_Shift[2]== 1){
11.               value1 = true;
12.             }else{
13.               value1 = false;
14.             }
15.           } else{
16.             if(roster[i][j].equalsIgnoreCase("P")){
17.                 Jml_Shift[0]++;
18.               } else if(roster[i][j].equalsIgnoreCase("S")){
19.                 Jml_Shift[1]++;
20.               } else if(roster[i][j].equalsIgnoreCase("M")){
21.                 Jml_Shift[2]++;
22.               }
23.
24.             if (Jml_Shift[0]== 3 && Jml_Shift[1]== 2 && Jml_Shift[2]== 1){
25.               value2 = true;
```

```
26.                     } else{
27.                         value2 = false;
28.                     }
29.                 }
30.             }
31.         }
```

Gambar 5.43 Kondisi pengechekan hard constraint unit Farmasi

Kode program pada Gambar 5.44 digunakan untuk melihat keluaran dari hasil *checking hard constraint*. Jika value1 dan value2 sama sama memiliki nilai *true* maka hasilnya merupakan true. Sedangkan jika nilai value1 dan dua tidak memiliki kesamaan maka bernilai *false*.

```
1.      System.out.println("Cek Constraint");
2.          System.out.println("Nilai value1 : " + value1);
3.          System.out.println("Nilai value2 : " + value2);
4.
5.          if (value1=value2) {
6.             value1=true;
7.          } else {
8.             value1=false;
9.          }
10.
11.       System.out.println(value1);
12.       return value1;
13.     }
```

Gambar 5.44 Menampilkan hasil cheking hard constraint unit Farmasi

5.2.5.2. Unit Nicu dan Ruang Bayi

Berikut kode program yang dibangun untuk memberikan batasan pada saat *generating* secara otomatis dan proses optimasi pada unit Niccu dan Ruang bayi.

Kode program yang dibangun pertama kali pada Gambar 5.45 adalah membentuk *method* atau fungsi yang bernama cekConstraint dengan nilai tipe balikan *boolean*. Melakukan deklarasi nilai *boolean* pertama kali merupakan *false*. Untuk melakukan pengechekan isi array jadwal maka dibangun perulangan sebanyak kolom dan perulangan sebanyak baris.

```
1.  public boolean cekConstraint(String [][] roster){
2.      boolean value1 = false;
3.      boolean value2 = false;
4.      for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.          String [] Shift ={"P","S", "M"};
6.          int [] Jml_Shift = {0,0,0};

 for (int i = 0; i < NumberOfEmployee; i++) {
```

Gambar 5.45 Inisiasi fungs boolean unit Nicu dan Ruang Bayi

Kode program pada Gambar 5.46 melakukan perintah untuk melihat komposisi shift per hari. Pengechekan hard constraint dibagi menjadi dua yaitu pada hari minggu dan ari kerja sehingaa kode program yang digunakan adalah *if else*. Sedangkan untuk menemukan jumlah masing-masing shift yang tercakup *hard constraint* digunakan fungsi *equalsignorecase* untuk mengambil nilai pada jadwal.

```
1.  f(roster[i][j].equalsIgnoreCase("P")){
2.                  Jml_Shift[0]++;
3.              } else if(roster[i][j].equalsIgnoreCase("S")){
4.                  Jml_Shift[1]++;
5.              } else if(roster[i][j].equalsIgnoreCase("M")){
6.                  Jml_Shift[2]++;
```

Gambar 5.46 Mengambil nilai shift pada jadwal Unit Nicu dan Bayi

```
1.  if (((Jml_Shift[0] >= 2) || (Jml_Shift[0] <= 4)) &&
2.                  ((Jml_Shift[1] >= 1) || (Jml_Shift[1] <= 2))
3.                  &&((Jml_Shift[2] >= 2) || (Jml_Shift[2] <= 3))){
4.              value1 = true;
5.          }else{
6.              value1 = false;
7.          }
```

Gambar 5.47 Kondisi pengechekan hard constraint unit Nicu dan Ruang Bayi

Kode program pada Gambar 5.47 melakukan perintah untuk melihat jumlah komposisi shift yang sesuai dengan *hard constraint.* Untuk melakukan pengechekan dilakukan

mengggunakan kodisi *if else* dengan ketentuan sama dengan *hard constraint*.

Kode program pada Gambar 5.48 digunakan untuk menampilakan hasil *checking hard cosntraint*.

```
1.  System.out.println("Cek Constraint");
2.      System.out.println("Nilai value1 : " + value1);
3.
4.    System.out.println(value1);
5.    return value1;
```

Gambar 5.48 Menampilkan hasil cheking hard cosntraint unit Nicu dan Ruang Bayi

5.2.5.3. Unit SIM dan RM

Berikut kode program yang dibangun untuk memberikan batasan pada saat *generating* secara otomatis dan proses optimasi pada unit SIM dan RM.

Kode program yang dibangun pertama kali pada Gambar 5.49 adalah membentuk *method* atau fungsi yang bernama cekConstraint dengan nilai tipe balikan *boolean*. Melakukan deklarasi nilai *boolean* pertama kali merupakan *false*. Untuk melakukan pengechekan isi array jadwal maka dibangun perulangan sebanyak kolom dan perulangan sebanyak baris.

```
1.  public boolean cekConstraint(String [][] roster){
2.      boolean value1 = false;
3.      boolean value2 = false;
4.
5.      for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.        String [] Shift ={"P","S","M"};
7.        int [] Jml_Shift = {0,0,0};
8.        for (int i = 0; i < NumberOfEmployee; i++) {
```

Gambar 5.49 Inisiasi fungs boolean unit SIM dan RM

Kode program Gambar 5.50 melakukan perintah untuk melihat komposisi shift per hari. Pengechekan hard constraint dibagi menjadi dua yaitu pada hari minggu dan ari kerja sehingaa kode program yang digunakan adalah *if else*. Sedangkan untuk menemukan jumlah masing-masing shift yang tercakup *hard*

constraint digunakan fungsi *equalsignorecase* untuk mengambil nilai pada jadwal.

```
1.  if(roster[i][j].equalsIgnoreCase("P")){
2.                  Jml_Shift[0]++;
3.                } else if(roster[i][j].equalsIgnoreCase("S")){
4.                  Jml_Shift[1]++;
5.                }else if(roster[i][j].equalsIgnoreCase("M")){
6.                  Jml_Shift[2]++;
7.                }
8.
9.              if (((Jml_Shift[0] >= 1) || (Jml_Shift[0] <= 2)) &&
10.                 ((Jml_Shift[1] >= 1) || (Jml_Shift[1] <= 2))
11.                 && (Jml_Shift[2] == 1)){
12.               value1 = true;
13.             }else{
14.               value1 = false;
15.             }
```

Gambar 5.50 Kondisi pengechekan hard constraint unit SIM dan RM

Kode program pada Gambar 5.51 digunakan untuk menampilakan hasil *checking hard cosntraint*.

```
1.  System.out.println("Cek Constraint");
2.      System.out.println("Nilai value1 : " + value1);
3.
4.    System.out.println(value1);
5.    return value1;
```

Gambar 5.51 Menampilkan hasil cheking hard constraint unit SIM dan RM

5.2.5.4. Unit IGD dan Rawat Jalan

Berikut kode program yang dibangun untuk memberikan batasan pada saat *generating* secara otomatis dan proses optimasi pada unit IGD dan Rawat Jalan.

Kode program yang dibangun pertama kali pada Gambar 5.52 adalah membentuk *method* atau fungsi yang bernama cekConstraint dengan nilai tipe balikan *boolean*. Melakukan deklarasi nilai *boolean* pertama kali merupakan *false*. Untuk melakukan pengechekan isi array jadwal maka dibangun perulangan sebanyak kolom dan perulangan sebanyak baris.

```
1.  public boolean cekConstraint(String [][] roster){
2.      boolean value1 = false;
3.      boolean value2 = false;
4.      for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.          String [] Shift ={"P","S", "M"};
6.          int [] Jml_Shift = {0,0,0};
7.          for (int i = 0; i < NumberOfEmployee; i++) {
```

Gambar 5.52 Inisiasi fungs boolean unit IGD dan Rawat Jalan

Kode program pada Gambar 5.53 melakukan perintah untuk melihat komposisi shift per hari. Pengechekan hard constraint dibagi menjadi dua yaitu pada hari minggu dan ari kerja sehingaa kode program yang digunakan adalah *if else*. Sedangkan untuk menemukan jumlah masing-masing shift yang tercakup *hard constraint* digunakan fungsi *equalsignorecase* untuk mengambil nilai pada jadwal.

```
1.  if(roster[i][j].equalsIgnoreCase("P")){
2.          Jml_Shift[0]++;
3.      } else if(roster[i][j].equalsIgnoreCase("S")){
4.          Jml_Shift[1]++;
5.      } else if(roster[i][j].equalsIgnoreCase("M")){
6.          Jml_Shift[2]++;
7.      }
```

Gambar 5.53 Mengambil nilai shift pada jadwal unit IGD dan Rawat Jalan

Kode program pada Gambar 5.54 melakukan perintah untuk melihat jumlah komposisi shift yang sesuai dengan *hard constraint.* Untuk melakukan pengechekan dilakukan mengggunakan kodisi *if else* dengan ketentuan sama dengan *hard constraint*.

```
1.      if (((Jml_Shift[0] >= 1)||(Jml_Shift[0] <= 2)) &&
2.
3.          ((Jml_Shift[1] >= 2) || (Jml_Shift[1] <= 4)) &&
4.          ((Jml_Shift[2] >= 1)||(Jml_Shift[2] <= 2))){
5.        value1 = true;
6.      }else{
7.        value1 = false;
8.      }
```

```
9.
```

Gambar 5.54 Kondisi pengechekan hard constraint unit IGD dan Rawat Jalan

Kode program pada Gambar 5.55 digunakan untuk menampilakan hasil *checking hard cosntraint*.

```
1.  System.out.println("Cek Constraint");
2.      System.out.println("Nilai value1 : " + value1);
3.      //System.out.println("Nilai value2 : " + value2);
4.
5.    System.out.println(value1);
```

Gambar 5.55 Menampilkan hasil cheking hard constraint unit IGD dan Rawat Jalan

5.2.5.5. Unit Gizi dan Cafe

Berikut kode program yang dibangun untuk memberikan batasan pada saat *generating* secara otomatis dan proses optimasi pada unit Gizi. Kode program dapat dilihat pada Gambar 5.56, Gambar 5.57, dan Gambar 5.58.

Kode program yang dibangun pertama kali pada Gambar 5.56 adalah membentuk *method* atau fungsi yang bernama cekConstraint dengan nilai tipe balikan *boolean*. Melakukan deklarasi nilai *boolean* pertama kali merupakan *false*. Untuk melakukan pengechekan isi array jadwal maka dibangun perulangan sebanyak kolom dan perulangan sebanyak baris.

```
1.  public boolean cekConstraint(String [][] roster){
2.      boolean value1= false;
3.      boolean value2= false;
4.
5.      for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.          String [] Shift ={"PS","M"};
7.          int [] Jml_Shift = {0,0,0};
8.          for (int i = 0; i < NumberOfEmployee; i++) {
```

Gambar 5.56 Inisiasi fungs boolean unit Gizi

Kode program pada Gambar 5.57 melakukan perintah untuk melihat komposisi shift per hari. Pengechekan hard constraint dibagi menjadi dua yaitu pada hari minggu dan hari kerja sehingaa kode program yang digunakan adalah *if else*. Sedangkan untuk menemukan jumlah masing-masing shift yang tercakup *hard constraint* digunakan fungsi *equalsignorecase* untuk mengambil nilai pada jadwal. Untuk melakukan perintah melihat jumlah komposisi shift yang sesuai dengan *hard constraint.* Untuk melakukan pengechekan dilakukan mengggunakan kodisi *if else* dengan ketentuan sama dengan *hard constraint*.

```
1.  if((j+FirstDay)%7 == 0){
2.                  if(roster[i][j].equalsIgnoreCase("PS")){
3.                      Jml_Shift[0]++;
4.                   } else if(roster[i][j].equalsIgnoreCase("M")){
5.                      Jml_Shift[1]++;
6.                   }
7.
8.                  if (Jml_Shift[0] == 2 && Jml_Shift[1]== 1){
9.                     value1 = true;
10.                 }else{
11.                    value1 = false;
12.                 }
13.             } else{
14.                  if(roster[i][j].equalsIgnoreCase("PS")){
15.                      Jml_Shift[0]++;
16.                   } else if(roster[i][j].equalsIgnoreCase("M")){
17.                      Jml_Shift[1]++;
18.                   }
19.                 if (Jml_Shift[1]== 1){
20.                    value2 = true;
21.                 } else{
22.                    value2 = false;
23.                 }
24.             }
```

Gambar 5.57 Kondisi pengechekan hard constraint unit Gizi dan Café

Kode program pada Gambar 5.58 digunakan untuk menampilakan hasil *checking hard cosntraint*.

```
1.  System.out.println("Cek Constraint");
2.      System.out.println("Nilai value1 : " + value1);
3.      System.out.println("Nilai value2 : " + value2);
4.
5.      if (value1=value2) {
6.         value1=true;
```

```
7.          } else {
8.             value1=false;
9.          }
10.
11.      System.out.println(value1);
```

Gambar 5.58 Menampilkan hasil cheking hard constraint unit Gizi dan Café

5.2.5.6. Ruang Operasi

Berikut kode program yang dibangun untuk memberikan batasan pada saat *generating* secara otomatis dan proses optimasi pada Ruang Operasi.

Kode program yang dibangun pertama kali pada Gambar 5.59 adalah membentuk *method* atau fungsi yang bernama *cekConstraint* dengan nilai tipe balikan *boolean*. Melakukan deklarasi nilai *boolean* pertama kali merupakan *false*. Untuk melakukan pengechekan isi array jadwal maka dibangun perulangan sebanyak kolom dan perulangan sebanyak baris.

```
1.    public boolean cekConstraint(String [][] roster){
2.          boolean value1 = false;
3.          boolean value2 = false;
4.
5.          for (int j = NumberOfDays; j
      <NumberOfDays+LengthOfMonth[newMonths]; j++) {
6.             String [] Shift ={"P","S"};
7.             int [] Jml_Shift = {0,0,0};
8.             for (int i = 0; i < NumberOfEmployee; i++) {
```

Gambar 5.59 Inisiasi fungs boolean Ruang Operasi

Kode program pada Gambar 5.60 melakukan perintah untuk melihat komposisi shift per hari. Pengechekan *hard constraint* menggunakan *if else*. Untuk melakukan perintah melihat jumlah komposisi shift yang sesuai dengan *hard constraint.* Sedangkan untuk mengambil nilai shift untuk satu hari menggunakan fungsi *equalsingnorecase* pada array jadwal. Untuk melakukan pengechekan dilakukan mengggunakan kodisi *if else* dengan ketentuan sama dengan *hard constraint*.

```
1.    if((j+FirstDay)%7 == 0){
2.             if(roster[i][j].equalsIgnoreCase("P")){
```

```
                    Jml_Shift[0]++;
                  } else if(roster[i][j].equalsIgnoreCase("S")){
                    Jml_Shift[1]++;
                  }

                if (Jml_Shift[0] == 4 && Jml_Shift[1]== 0 ){
                  value1 = true;
                } else{
                  value1 = false;
                }

              } else{
                if(roster[i][j].equalsIgnoreCase("P")){
                    Jml_Shift[0]++;
                  } else if(roster[i][j].equalsIgnoreCase("S")){
                    Jml_Shift[1]++;
                  }

                if (((Jml_Shift[0] >= 2) || (Jml_Shift[0] <= 4)) &&
                      ((Jml_Shift[1] >= 2) || (Jml_Shift[1] <= 3))){
                  value2 = true;
                }else{
                  value2 = false;
                }
              }
```

Gambar 5.60 Kondisi pengechekan hard constraint Ruang Operasi

Kode program pada Gambar 5.61 digunakan untuk menampilakan hasil *checking hard cosntraint*.

```
System.out.println("Cek Constraint");
      System.out.println("Nilai value1 : " + value1);
      System.out.println("Nilai value2 : " + value2);

      if (value1=value2) {
        value1=true;
      } else {
        value1=false;
      }

    System.out.println(value1);
```

Gambar 5.61 Menampilkan hasil cheking hard constraint Ruang Operasi

5.3. Implementasi Algoritma

Pada sub bab ini dilakukan optimasi feasible schedule dengan menggunakan algoritma *tabu serach based hyper-heuristics*.

5.3.1. Implementasi Algoritma Hyper-Heuristics

Pada sub-sub bab ini dilakukan pembuatan *heuristics* yang akan dipilih algoritma tabu serach dalam melakukan optimasi pendawalan. Kode program yang dibangun untuk merancang heuristics terdiri dari:

a. Swap

Heuristics swap digunakan untuk menukar alokasi libur. Penukaran alokasi libur dilakukan secara random pada slot jadwal pada satu hari. Syarat *swap* atau penukaran akan dapat diterima jika tidak melanggar *hard constraint* harian yang telah ditetapkan. Kode program low level heuristics dapat dilihat pada Gambar 5.62.

```
1.  public void Swap(){
2.        List<Integer>  posisiLHari = new ArrayList<>();
3.        List<Integer>  posisiLEmployee = new ArrayList<>();
4.        for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
5.            for (int i = 0 ; i <NumberOfEmployee; i++) {
6.              if(roster[i][j].equalsIgnoreCase("L")){
7.                 posisiLHari.add(new Integer(j));
8.                 posisiLEmployee.add(new Integer(i));
9.              }
10.           }
11.       }
12.       int memilihHari = rand.nextInt(posisiLHari.size());
13.       String temp =
    roster[posisiLEmployee.get(memilihHari)][posisiLHari.get(memilihHari
    )];
14.       int memilihEmployee;
15.
16.       do {
17.          memilihEmployee = rand.nextInt(NumberOfEmployee);
18.       }while(memilihEmployee == posisiLEmployee.get(memilihHari));
19.
    roster[posisiLEmployee.get(memilihHari)][posisiLHari.get(memilihHari
    )]
20.           =
    roster[posisiLEmployee.get(memilihEmployee)][posisiLHari.get(memili
    hHari)];
21.
    roster[posisiLEmployee.get(memilihEmployee)][posisiLHari.get(memili
    hHari)]= temp;
```

```
22.     }
```

Gambar 5.62 Fungsi low level heuristics Swap()

b. Move

Heuristics digunakan untuk menambahkan jumlah libur yang kurang dengan jatah libur yang telah ditetapkan dalam bulan desember. Penambahan libur tersebut bertujuan dapat memaksimalkan nilai *jain fairness index* sehingga jadwal yang dihasilkan memiliki tingkat keadilan antar staf yang baik. Kode program *low level heuristics* dapat di lihat pada Gambar 5.63.

```
1.  public void move(){
2.      int  JmlLibur;
3.      int selisih;
4.      int JumlahLibur_seharusnya = (LengthOfMonth[newMonths]/7);
5.       if(((NumberOfDays + (FirstDay - 1)) % 7 ) >=4){
6.          JumlahLibur_seharusnya ++;
7.       }
8.      for (int i = 0 ; i <NumberOfEmployee; i++) {
9.         JmlLibur=0;
10.        selisih=0;
11.        List<Integer>  posisiMS = new ArrayList<>();
12.          for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
13.            if(roster[i][j].equalsIgnoreCase("L")){
14.                JmlLibur++;
15.              }
16.            if (roster [i][j].equalsIgnoreCase("MS")){
17.              posisiMS.add(new Integer(j));
18.             }
19.          }
20.          selisih = JumlahLibur_seharusnya-JmlLibur;
21.          if (selisih >0 && posisiMS.size()>0){
22.          for (int k = 0; k < selisih; k++) {
23.            int IndexMSygDiganti = rand.nextInt(posisiMS.size());
24.              roster[i][posisiMS.get(IndexMSygDiganti)]= "L";
25.              }
26.          }
27.      }
28.   }
```

Gambar 5.63 Fungsi low level heuristics Move()

5.3.2. Pemilihan Solusi dengan Menggunakan Algoritma Hill Climbing

Pada sub-sub bab ini dilakukan pembuatan algoritma *hill climbing* yang berguna untuk melakukan pemilihan solusi heuristics terbaik dalam melakukan optimasi pendawalan. Algoritma *hill climbing* merupakan algoritma dasar yang akan digunakan untuk membangun algoritma *tabu search*. Secara logika algoritma *hill climbing* akan bekerja mencari solusi terbaik secara terus menerus dengan membandingkan nilai terbaik saat ini dengan nilai terbaik yang baru. Kode program algoritma *hill climbing* dapat dilihat pada Gambar 5.64.

```
1.  public void HillClimbing(int NumbIt){
2.      String []][] bestSolution = new String
    [NumberOfEmployee][LengthOfMonth[newMonths]];
3.      bestJFIcost = curJFIcost = JFi(roster);
4.      String []][] Solpertama = new String
    [NumberOfEmployee][LengthOfMonth[newMonths]];
5.
6.      for (int i = 0; i < NumbIt; i++) {
7.        System.out.printf("%2d, %5f, %5f \n", i,curJFIcost, bestJFIcost );
8.      Solpertama = roster.clone(); //this is the best Solution So Far
9.            if (Math.random() <0.5) {
10.             move();
11.             //System.out.println("pilih move");
12.           } else {
13.             Swap();
14.             //System.out.println("pilih swap");
15.           }
16.
17.       curJFIcost= JFi(roster);
18.
19.       if (curJFIcost > bestJFIcost) {
20.         bestJFIcost = curJFIcost;
21.         System.arraycopy(roster, 0, bestSolution, 0, bestSolution.length);
22.       } else{
23.         roster = Solpertama.clone();
24.         System.arraycopy(roster, 0, bestSolution, 0, bestSolution.length);
25.       }
26.     }
27.     System.out.println("BestJFI, " + bestJFIcost );
28.
29.     for (int i = 0 ; i <NumberOfEmployee; i++) {
30.     for (int j = NumberOfDays; j
    <NumberOfDays+LengthOfMonth[newMonths]; j++) {
31.         System.out.printf("%6s,", bestSolution[i][j]);
32.         }
33.         System.out.println(" ");
34.     }
```

```
35.     }
```

Gambar 5.64 Kode program algoritma hill climbing

5.3.3. Pemilihan Solusi dengan Menggunakan Tabu Search

Pada sub-sub bab ini dilakukan pembuatan algoritma *tabu search* yang berguna untuk melakukan pemilihan solusi heuristics terbaik dalam melakukan optimasi pendawalan. Kode program *tabu serach* dibangun berdasarkan rancangan algoritma dan *pseudocode* pada sub bab 4.5.

Kode program tabu search yang dibangun dapat dilihat pada Gambar 5.65.

```
1.  public void tabusearch(int NumbIteration){
2.      String [][] bestSolution = new String
    [NumberOfEmployee][LengthOfMonth[newMonths]];
3.      PriorityQueue<String> queue=new PriorityQueue<String>();
4.      String [][] Solpertama = new String
    [NumberOfEmployee][LengthOfMonth[newMonths]];
5.
6.      bestJFIcost = curJFIcost = JFi(roster);
7.      for (int i = 0; i < NumbIteration; i++) {
8.         System.out.printf("%2d, %5f, %5f \n", i,curJFIcost, bestJFIcost
    );
9.      String selectedLLH;
10.     Solpertama = roster.clone(); //this is the best Solution So Far
11.             //CurrentSol = cari solusi dengan menggunakan
    hyperheuristics
12.         if (queue.isEmpty()){
13.            if (Math.random() <0.5) {
14.               move();
15.               selectedLLH = "M";
16.               //System.out.println("pilih move");
17.            } else {
18.               Swap();
19.               selectedLLH = "S";
20.               //System.out.println("pilih swap");
21.            }
22.         }else{
23.            if (queue.peek().equalsIgnoreCase("S")){
24.               move();
25.               selectedLLH = "M";
26.            }else {
```

```
27.                     Swap();
28.                     selectedLLH = "S";
29.                   }
30.               }
31.
32.               curJFIcost= JFi(roster);
33.              if (curJFIcost > bestJFIcost) {
34.                 bestJFIcost = curJFIcost;
35.              } else{
36.                 if (!queue.isEmpty()){
37.                    queue.poll();
38.                 }
39.                 queue.add(selectedLLH);
40.                 roster = Solpertama.clone();
41.              }
42.           }
43.           System.out.println("BestJFI, " + bestJFIcost );
44.           for (int i = 0 ; i <NumberOfEmployee; i++) {
45.           for (int j = NumberOfDays; j
      <NumberOfDays+LengthOfMonth[newMonths]; j++) {
46.                 System.out.printf("%6s,", roster[i][j]);
47.                 }
48.                  System.out.println(" ");
49.           }
50.       }
```

Gambar 5.65 Aloritma tabu search

(Halaman ini sengaja dikosongkan)

BAB VI
HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan mengenai proses uji coba dan analisis terhadap hasil yang diperoleh dari proses implementasi Algoritma *Tabu Search Based Hyper-heuristics*.

6.1.Validasi *Feasible Schedule*

Pada sub-bab ini dilakukan pengechekan hasil jadwal yang dibuat secara otomatis berdasarkan *hard constraint* yang ada. Proses validasi digunakan untuk memastikan kebenaran kode program yang telah dibangun. Hasil *generating* jadwal yang *feasible* terlampir pada lampiran B. Pada sub-bab ini juga dilakukan pengechekan hasil optimasi jadwal dengan menggunakan algoritma *tabu serach based hyper heuristics* yang dibuat secara otomatis berdasarkan *hard constraint*. Hasil optimasi jadwal dapat dilihat pada lampiran D.

Tabel 6.1 berikut merupakan hasil validasi *hard constraint* untuk *feasible schedule* dan hasil optimasi unituk masing-masing unit.

Tabel 6.1Validasi Hasil *Feasible Schedule*

No	Unit	Kode Hard Constraint	Hasil Generating	Hasil Optimasi
1	Farmasi	AI	Terpenuhi	Terpenuhi
		A2	Terpenuhi	Terpenuhi
		A3	Terpenuhi	Terpenuhi
		A4	Terpenuhi	Terpenuhi
2	NICU Ruang Bayi	B1	Terpenuhi	Terpenuhi
		B2	Terpenuhi	Terpenuhi
		B3	Terpenuhi	Terpenuhi
3	SIM & RM	C1	Terpenuhi	Terpenuhi
		C2	Terpenuhi	Terpenuhi

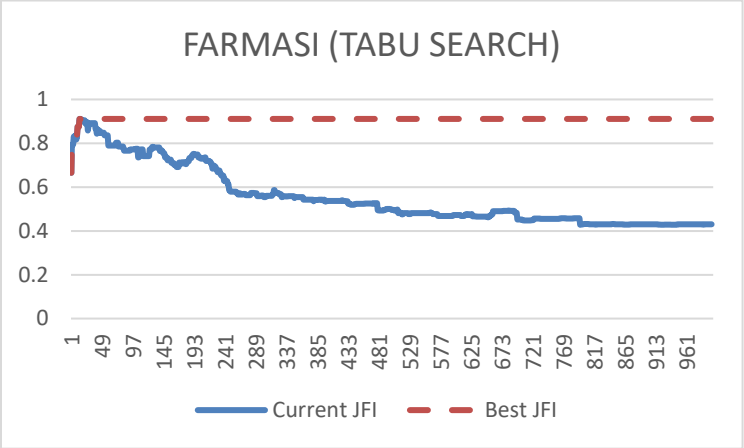
No	Unit	Kode Hard Constraint	Hasil Generating	Hasil Optimasi
		C3	Terpenuhi	Terpenuhi
4	IGD dan POLI	D1	Terpenuhi	Terpenuhi
		D2	Terpenuhi	Terpenuhi
		D3	Terpenuhi	Terpenuhi
5	Gizi	E1	Terpenuhi	Terpenuhi
		E2	Terpenuhi	Terpenuhi
6	OK	F1	Terpenuhi	Terpenuhi
		F2	Terpenuhi	Terpenuhi
		F3	Terpenuhi	Terpenuhi

Berdasarkan Tabel 6.1 diatas dapat disimpulkan seluruh *costrtraint* dari setiap unit dapat terpenuhi baik pada saat *generating* jadwal secara otomatis dan pada saat optimasi.

6.2. Hasil Implementasi Penjadwlan Staf Unit Farmasi

Hasil implementasi penjadwalan staf unit Farmasi dengan menggunakan *tabu search based hyper heuristics* dapat dilihat pada gambar lampiran D. Hasil optimasi penjadwalan dipilih berdasakan nilai paling optimum yang dihasilkan dari kode program yang dibangun. Sifat *hyper heuristics* yang kemunculanya tidak pasti maka penulis melakukan eksplorasi pada dua algoritma yaitu *hill climbing* dan *tabu search* untuk menentukan hasil yang paling optimal.

Langkah pertama yang dilakukan penulis adalah melakukan optimasi dengan menggunakan algoritma *tabu search* pada jadwal unit farmasi dengan parameter masukan iterasi sebanyak seribu.

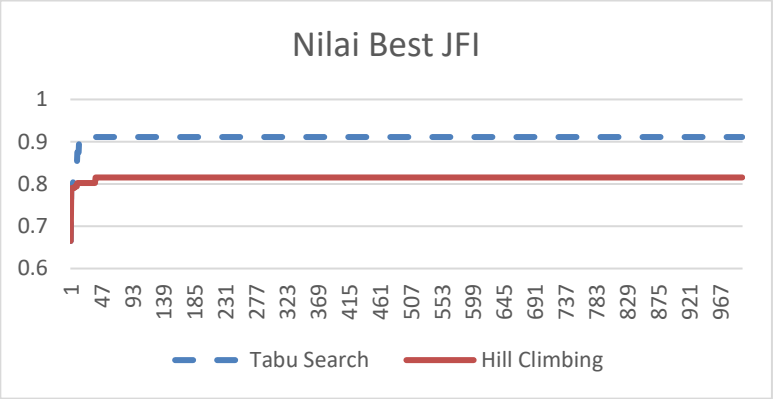


Gambar 6.1 Hasil 1000 iterasi algoritma tabu search pada unit Farmasi

Berdasarkan Gambar 6.1 menyebutkan semakin tinggi tingkat itersi pada algoritma *tabu search* menghasilkan cost JFI yang semakin turun secara eksponential. Nilai JFI naik mulai iterasi pertama sampai iterasi rentan antara 50 sampai 100. Hal tersebut dapat disimpulkan bahwa performa algoritma *tabu search* dengan data jadwal farmasi memiliki kemampuan cukup cepat untuk menemukan solusi terbaik.

Langkah kedua adalah membandingkan *best* JFI yang didapatkan dari algoritma *tabu search* dan algoritma *hill climbing*. Hasil best JFI didapatkan dengan parameter iterasi sebanyak seribu dengan satu kali eksekusi program.

Berdasarkan *line chart* pada Gambar 6.2 algoritma *tabu search* menghasilkan best JFI lebih tinggi dari algoritma *hill climbing*. Algoritma *tabu search* dan *hill climbing* memiliki kesamaan dalam kecepatan menemukan nilai best JFI yaitu pada iterasi ke 100.



Gambar 6.2 Nilai best JFI pada unit Farmasi

Berdasarkan hasil percobaan pada Gambar 6.1 dan Gambar 6.2 maka dapat digunakan sebagai acuan berapa kali iterasi yang baik untuk melakukan optimasi. Sehingaa penulis berasumsikan untuk melakukan percobaan iterasi sebanyak seratus kali iterasi dan dua ratus iterasi dengan percobaan optimasi sebanyak dua puluh satu kali. Tabel 6.2 dibawah ini merupakan hasil uji coba optimasi dengan menggunakan algoritma *tabu search* dan *hill climbing*.

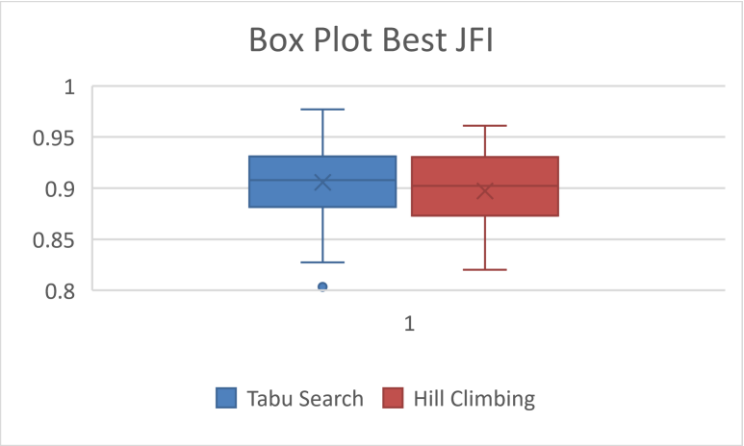
Tabel 6.2 Hasil uji coba optimasi unit Farmasi

NO	Algoritma	Jumlah Iterasi	Nilai	
1	Tabu Search	100	Max	0.976879
			Min	0.856164
			Avg	0.905252
		200	Max	0.968478
			Min	0.803342
			Avg	0.905989
2	Hill Climbing	100	Max	0.960908
			Min	0.82021
			Avg	0.882481
		200	Max	0.957274

NO	Algoritma	Jumlah Iterasi	Nilai	
			Min	0.859697
			Avg	0.911613

Berdasarkan infromasi pada Tabel 6.2, nilai *jains fairness index* tertinggi didapatkan dengan menggunakan algoritma *tabu search* dengan jumlah iterasi sebanyak seratus kali yaitu sebesar 0.976879. Dan hasil *jains fairness index* terkecil juga didapatkan pada optimasi dengan menggunakan algoritma *tabu search* dengan iterasi sebanyak dua ratus kali yaitu sebesar 0.803342. Rincian hasil secara lengkap terdapat pada lampiran C.

Untuk mengetahui persebaran hasil optimasi kedua algoritma berikut ini merupakan gambar Box plot pada Gambar 6.3. berdasarkan box plot, kedua algoritma memiliki hasil yang sama yaitu memiliki rata rata pada angka 0.9.



Gambar 6.3 Box plot *best* JFI unit Farmasi

Perbandingan nilai JFI atau *jain fairness index* digunakan untuk mengetahui nilai JFI yang paling maksimal. Perbandingan JFI

dilihat dari jadwal manual bulan November 2017, jadwal otomatis Desember 2017, dan jadwal Desember 2017 hasil optimasi.

Tabel 6.3 Perbandingan nilai JFI Unit Farmasi

Unit farmasi		
No	**Jadwal pada Bulan ke-**	**JFI**
1	Jadwal Manual November 2017	0.80798005
2	Jadwal Otomatis Desember 2017	0.66515837
3	Jadwal Desember 2017 (hasil optimasi)	0.976879

Berdasarkan Tabel 6.3 menyebutkan bahwa nilia *jain fairness index* jadwal manual November 2017 lebih tinggi dari jadwal hasil generating otomatis bulan Desember 2017. Dan hasil optimasi pada jadwal bulan Desember 2017 memiliki nilai JFI sebesar 0.9768, meningkat sebesar 47% dari nilai JFI semula sebelum jadwal dioptmasikan. Hal tersebut menyatakan tingkat keadilan yang sangat tinggi dikarenakan nilai mendekati satu.

Dari hasil optimasi penjadwalan unit Farmasi sesuai dengan lampiran D, pada Tabel 6.4 dibawah ini dilakukan perhitungan komposisi shift setiap staf yang dijadwalkan. Berdasarkan Tabel 6.4 dapat dilihat bahwa optimasi penjadwalan dapat memberikan nilai *fairness* kepada pembagian hari libur sesuai dengan pembobotan akan tetapi kurang bagus dalam pemerataan jam kerja.

Tabel 6.4 Jumlah shift setiap staf unit Farmasi

Id Staf	Tipe Shift								Total Jam Kerja
	P	S	MS	L	CT	M	LM	Total	
101	25	2	0	4	0	0	0	31	189
102	0	23	0	7	0	1	0	31	168
103	6	7	3	5	0	7	3	31	182
104	8	6	2	8	0	5	2	31	161

Id Staf	Tipe Shift								Total Jam Kerja
	P	S	MS	L	CT	M	LM	Total	
105	7	10	0	5	0	6	3	31	182
106	6	6	3	7	0	6	3	31	168
107	8	7	0	6	0	5	5	31	175
108	23	1	0	6	0	1	0	31	175

Pada Tabel 6.5 akan dijabarkan pembagagian hari libur antar staf pada unit. Pembagian hari libur akan dihitung berdasarkan jumlah hari libur pada hari sabtu, jumlah libur pada hari minggu dan jumlah libur pada hari kerja atau *weekdays*.

Tabel 6.5 Jumlah hari libur setiap staf unit Farmasi

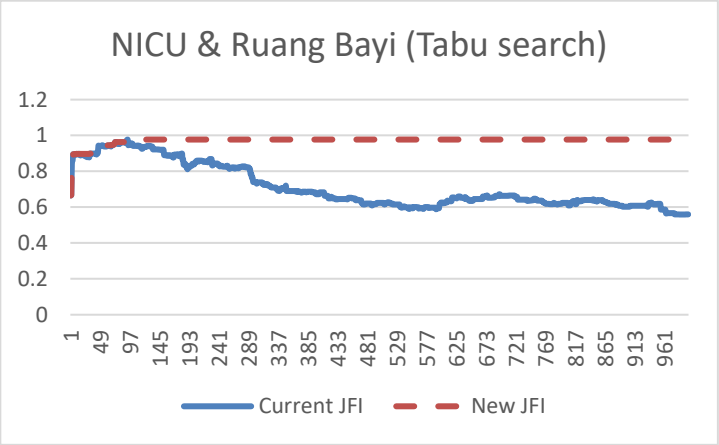
Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
101	3	0	0	1	0	0	0	4
102	3	0	1	0	0	1	2	7
103	2	0	0	1	1	1	0	5
104	2	1	1	3	0	0	1	8
105	2	1	1	0	0	1	0	5
106	1	1	2	0	1	1	1	7
107	1	1	0	0	2	1	1	6
108	3	1	0	0	1	0	1	6

Dimana M merupakan hari mingu, Sb merupakan hari sabtu, Sn merupakan hari senin, Sl merupakan hari selasa, R merupakan hari rabo, K merupakan hari kamis, dan J merupakan hari Jum’at.

6.3. Hasil Implementasi Penjadwalan Staf Unit NICU dan Ruang Bayi

Hasil implementasi penjadwalan staf unit Nicu dan Ruang Bayi dengan menggunakan *tabu search based hyper heuristics* dapat dilihat pada gambar lampiran D. Hasil optimasi penjadwalan dipilih berdasakan nilai paling optimum yang dihasilkan dari kode program yang dibangun.

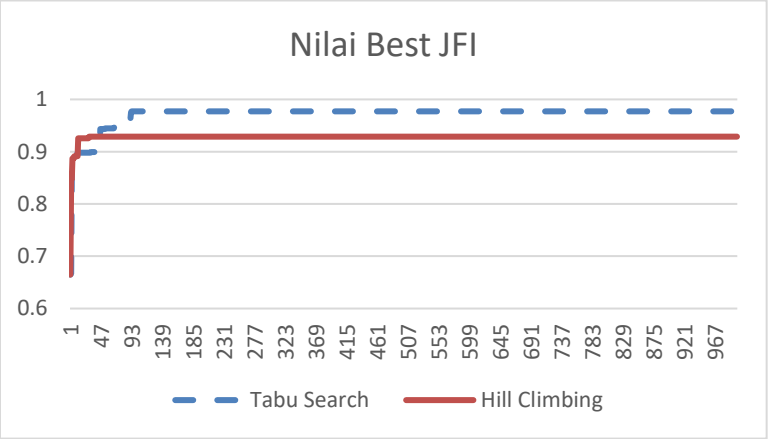
Langkah pertama yang dilakukan penulis adalah melakukan optimasi dengan menggunakan algoritma *tabu search* pada jadwal unit Nicu dan Ruang Bayi dengan parameter masukan iterasi sebanyak seribu.



Gambar 6.4 Hasil 1000 iterasi algoritma tabu search pada unit Nicu dan Ruang Bayi

Berdasarkan Gambar 6.4 menyebutkan semakin tinggi tingkat itersi pada algoritma *tabu search* menghasilkan *cost* JFI yang semakin turun secara eksponential. Nilai JFI naik mulai iterasi pertama sampai iterasi ke 100 sampai 200. Hal tersebut dapat disimpulkan bahwa performa algoritma *tabu search* dengan data jadwal unit Nicu dan Ruang Bayi memiliki kemampuan cukup cepat untuk menemukan solusi terbaik.

Langkah kedua adalah membandingkan *best* JFI yang didapatkan dari algoritma *tabu search* dan algoritma *hill climbing*. Hasil best JFI didapatkan dengan parameter iterasi sebanyak seribu dengan satu kali eksekusi program.



Gambar 6.5 Perbandingan nilai JFI terbaik unit Nicu dan Ruang Bayi

Berdasarkan Gambar 6.5 algoritma *tabu search* menghasilkan best JFI lebih tinggi dari algoritma *hill climbing*. Algoritma *hill climbing* memiliki kecepatan untuk mendapatkan solusi lebih cepat yaitu pada iterasi ke 50 dibandingkan dengan algoritma *tabu search* yang memiliki kecepatan menemukan nilai *best* JFI yaitu pada iterasi ke 130.

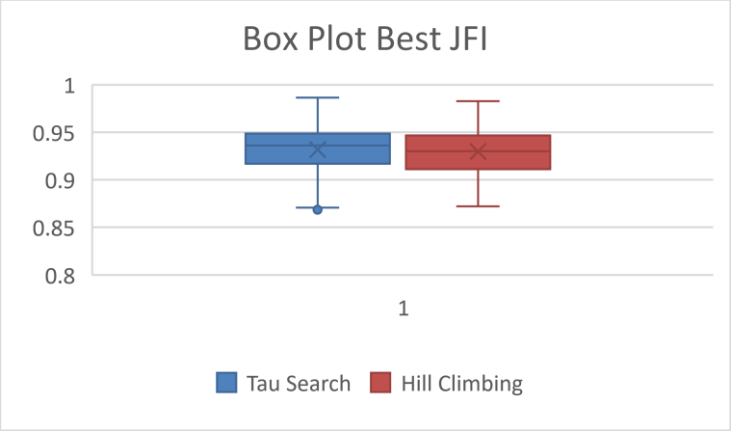
Berdasarkan hasil percobaan pada Gambar 6.4 dan Gambar 6.5 maka dapat digunakan sebagai acuan berapa kali iterasi yang baik untuk melakukan optimasi. Sehingaa penulis berasumsikan untuk melakukan percobaan iterasi sebanyak seratus kali iterasi dan dua ratus iterasi dengan percobaan optimasi sebanyak dua puluh satu kali. Tabel 6.6 dibawah ini merupakan hasil uji coba optimasi dengan menggunakan algoritma *tabu search* dan *hill climbing*.

Tabel 6.6 Hasil uji coba optimasi unit Nicu dan Ruang Bayi

NO	Algoritma	Jumlah Iterasi	Nilai	
1	Tabu Search	100	Max	0.986343
			Min	0.868682
			Avg	0.930831
		200	Max	0.986343
			Min	0.868682
			Avg	0.932668
2	Hill Climbing	100	Max	0.98263
			Min	0.871989
			Avg	0.932436
		200	Max	0.970232
			Min	0.891277
			Avg	0.927069

Berdasarkan infromasi pada tabel Tabel 6.6, nilai *jains fairness index* tertinggi didapatkan dengan menggunakan algoritma *tabu search* dengan jumlah iterasi sebanyak seratus kali yaitu sebesar 0.986343. Dan hasil *jains fairness index* terkecil juga didapatkan pada optimasi dengan menggunakan algoritma *tabu search* dengan iterasi sebanyak dua ratus kali yaitu sebesar 0.868682. Rincian hasil secara lengkap terdapat pada lampiran C.

Untuk mengetahui persebaran hasil optimasi kedua algoritma berikut ini merupakan gambar Box plot pada Gambar 6.6. berdasarkan box plot, kedua algoritma memiliki hasil yang sama yaitu memiliki rata rata pada angka 0.9. Nilai JFI maksimal didapatkan dari algoritma *tabu search* dengan iterasi sebanyak seratu kali dan nilai JFI terendah didapatkan dengan algoritma *tabu search* dengan jumlah iterasi sebanyak dua ratus kali.



Gambar 6.6 Box plot best JFI unit Nicu dan Ruang Bayi

Perbandingan nilai JFI atau *jain fairness index* digunakan untuk mengetahui nilai JFI yang paling maksimal. Perbandingan JFI dilihat dari jadwal manual bulan November 2017, jadwal otomatis Desember 2017, dan jadwal Desember 2017 hasil optimasi.

Tabel 6.7 Perbandingan nilai JFI Unit Nicu dan Ruang Bayi

Unit Niccu dan Bayi		
No	**Jadwal pada Bulan ke-**	**JFI**
1	Jadwal Manual November 2017	0.898027776
2	Jadwal Otomatis Desember 2017	0.664772727
3	Jadwal Desember 2017 (hasil optimasi)	0.986343

Berdasarkan Tabel 6.11 menyebutkan bahwa nilia *jain fairness index* jadwal manual November 2017 lebih tinggi dari jadwal hasil *generating* otomatis bulan Desember 2017. Dan hasil optimasi pada jadwal bulan Desember 2017 memiliki nilai JFI sebesar 0.986343, meningkat sebesar 48% dari nilai JFI semula sebelum jadwal dioptmasikan. Hal tersebut menyatakan tingkat keadilan yang sangat tinggi dikarenakan nilai mendekati satu.

Dari hasil optimasi penjadwalan unit Nicu dan Ruang Bayi sesuai dengan lampiran D, pada Tabel 6.8 dibawah ini dilakukan perhitungan komposisi shift setiap staf yang dijadwalkan. Berdasarkan Tabel 6.8 dapat dilihat bahwa optimasi penjadwalan dapat memberikan nilai *fairness* kepada pembagian hari libur sesuai dengan pembobotan akan tetapi kurang bagus dalam pemerataan jam kerja.

Tabel 6.8 Jumlah shift setiap staf unit Nicu dan Ruang Bayi

Id Staf	Tipe Shift								Total Jam Kerja
	P	S	MS	L	CT	M	LM	Total	
201	25	1	0	4	0	0	1	31	189
202	7	6	4	5	0	7	2	31	182
203	7	6	2	5	0	7	4	31	182
204	4	7	4	7	0	6	3	31	168
205	6	7	3	8	0	4	3	31	161
206	6	5	2	6	0	8	4	31	175
207	5	4	3	8	0	8	3	31	161
208	7	4	4	7	0	6	3	31	168
209	6	7	3	9	0	4	2	31	154
210	8	7	3	5	0	5	3	31	182
211	6	7	2	6	0	6	4	31	175
212	7	6	3	5	0	7	3	31	182
213	6	7	3	5	0	7	3	31	182

Pada Tabel 6.9 akan dijabarkan pembagagian hari libur antar staf pada unit. Pembagian hari libur akan dihitung berdasarkan jumlah hari libur pada hari sabtu, jumlah libur pada hari minggu dan jumlah libur pada hari kerja atau *weekdays*.

Tabel 6.9 Jumlah hari libur setiap staf unit Nicu dan Ruang Bayi

Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
201	2	0	0	1	1	0	0	4
202	2	1	1	0	0	1	0	5
203	2	0	0	1	0	1	1	5
204	1	0	0	0	1	2	3	7
205	0	1	2	2	1	1	1	8
206	1	1	1	2	0	0	1	6
207	1	2	2	1	1	1	0	8
208	1	2	1	1	1	0	1	7
209	1	0	1	1	2	2	2	9
210	1	2	0	1	1	0	0	5
211	2	1	1	0	0	1	1	6
212	2	1	0	0	0	1	1	5
213	1	1	1	0	1	0	1	5

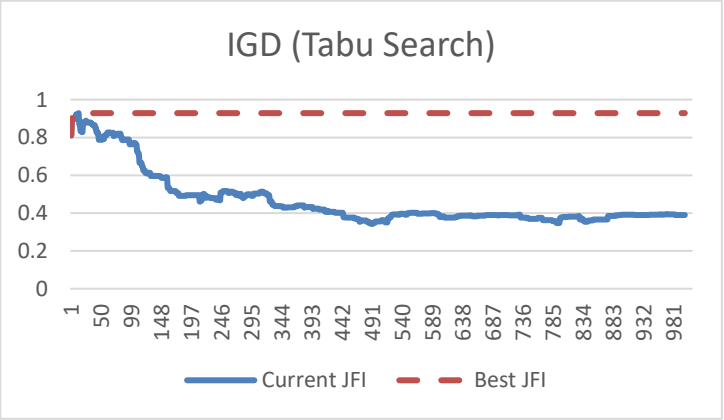
Dimana M merupakan hari mingu, Sb merupakan hari sabtu, Sn merupakan hari senin, Sl merupakan hari selasa, R merupakan hari rabo, K merupakan hari kamis, dan J merupakan hari Jum'at.

6.4. Hasil Implementasi Penjadwalan Staf Unit IGD dan Rawat Jalan

Hasil implementasi penjadwalan staf unit IGD dan Rawat jalan dengan menggunakan *tabu search based hyper heuristics* dapat dilihat pada gambar lampiran D. Hasil optimasi penjadwalan dipilih berdasakan nilai paling optimum yang dihasilkan dari kode program yang dibangun. Optimasi akan dilakukan dengan menggunakan dua algoritma yaitu *hill climbing* dan *tabu search*. Untuk menemukan hasil yang paling optimum maka penulis melakukan beberapa kali percobaan dengan berbagai macam parameter iterasi yang berbeda beda.

Langkah pertama yang dilakukan penulis adalah melakukan optimasi dengan menggunakan algoritma *tabu search* pada jadwal unit IGD dan Rawat jalan dengan parameter masukan iterasi sebanyak seribu.

Berdasarkan Gambar 6.7 di bawah menyebutkan semakin tinggi tingkat itersi pada algoritma *tabu search* menghasilkan *cost* JFI yang semakin turun secara eksponential. Nilai JFI naik mulai iterasi pertama sampai iterasi ke 100 sampai 200. Hal tersebut dapat diketahui bahwa performa algoritma *tabu search* dengan data jadwal unit IGD dan Rawat Jalan memiliki kemampuan cukup cepat untuk menemukan solusi terbaik.

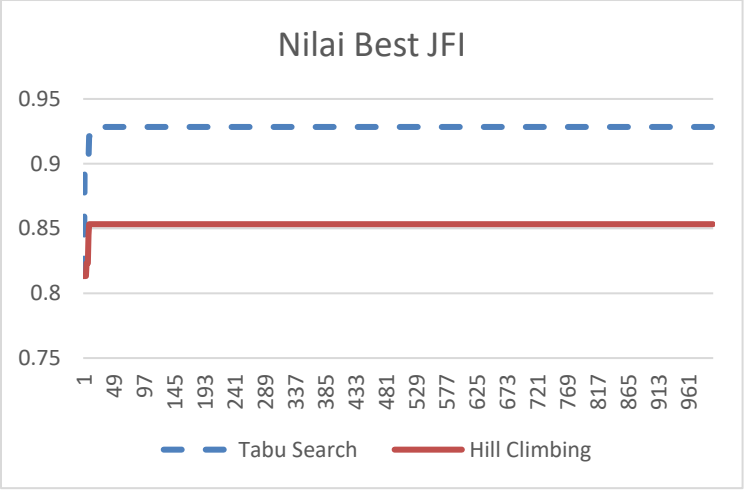


Gambar 6.7 Hasil 1000 iterasi algoritma tabu search pada unit IGD dan Rawat Jalan

Langkah kedua adalah membandingkan *best* JFI yang didapatkan dari algoritma *tabu search* dan algoritma *hill climbing*. Hasil *best* JFI didapatkan dengan parameter iterasi sebanyak seribu dengan satu kali eksekusi program.

Berdasarkan Gambar 6.8 algoritma *tabu search* menghasilkan best JFI lebih tinggi dari algoritma *hill climbing*. Algoritma *tabu search* dan *hill climbing* sama-sama memiliki kecepatan

untuk mendapatkan solusi lebih cepat yaitu pada iterasi ke lima puluh sampai seratus.



Gambar 6.8 Perbandingan nilai JFI terbaik unit IGD dan rawat Jalan

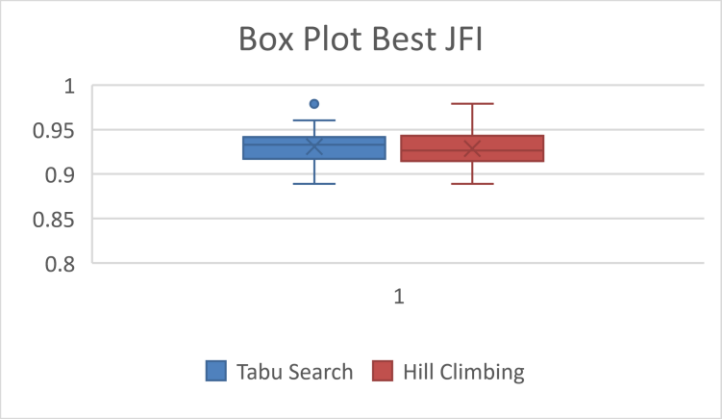
Berdasarkan hasil percobaan pada Gambar 6.7 dan Gambar 6.8 maka dapat digunakan sebagai acuan berapa kali iterasi yang baik untuk melakukan optimasi. Sehingaa penulis berasumsikan untuk melakukan percobaan iterasi sebanyak seratus kali iterasi dan dua ratus iterasi dengan percobaan optimasi sebanyak dua puluh satu kali. Tabel 6.10 dibawah ini merupakan hasil uji coba optimasi dengan menggunakan algoritma *tabu search* dan *hill climbing*.

Tabel 6.10 Hasil uji coba optimasi unit IGD dan Rawat Jalan

NO	Algoritma	Jumlah Iterasi	Nilai	
1	Tabu Search	100	Max	0.960227
			Min	0.897989
			Avg	0.932587
		200	Max	0.978827
			Min	0.888889

NO	Algoritma	Jumlah Iterasi	Nilai	
			Avg	0.928741
2	Hill Climbing	100	Max	0.96305
			Min	0.888889
			Avg	0.928116
		200	Max	0.978827
			Min	0.888889
			Avg	0.928741

Berdasarkan infromasi pada tabel 6.4.1, nilai *jains fairness index* paling tinggi didapatkan dengan menggunakan algoritma *tabu search* dan *hill climbing* dengan jumlah iterasi sebanyak dua ratus kali, yaitu sebesar 0.978827. Dan hasil *jains fairness index* terkecil juga didapatkan pada optimasi dengan menggunakan algoritma *tabu search* dengan iterasi sebanyak dua ratus kali dan dengan menggunakan algoritma *hill climbing* dengan iterasi sebanyak dua ratu dan seratus yaitu sebesar 0.888889. Rincian hasil secara lengkap terdapat pada lampiran C.



Gambar 6.9 Box plot unit IGD dan Rawat Jalan

Untuk mengetahui persebaran hasil optimasi kedua algoritma berikut ini merupakan gambar Box plot pada Gambar 6.9. berdasarkan box plot, kedua algoritma memiliki hasil yang sama yaitu memiliki rata rata pada angka 0.9.

Nilai JFI maksimal didapatkan dari algoritma *tabu search* dan algoritma *hill climbing* dengan iterasi sebanyak dua ratus kali dan nilai JFI terendah didapatkan dengan algoritma *tabu search* dengan jumlah iterasi sebanyak dua ratus kali dan algoritma *hill climbing* dengan iterasi sebanyak seratu kali dan dua ratus kali.

Perbandingan nilai JFI atau *jain fairness index* digunakan untuk mengetahui nilai JFI yang paling maksimal. Perbandingan JFI dilihat dari jadwal manual bulan November 2017, jadwal otomatis Desember 2017, dan jadwal Desember 2017 hasil optimasi.

Tabel 6.11 Perbandingan nilai JFI Unit IGD dan Rawat Jalan

Unit IGD		
No	**Jadwal pada Bulan ke-**	**JFI**
1	Jadwal Manual November 2017	0.875473485
2	Jadwal Otomatis Desember 2017	0.8132780082
3	Jadwal Desember 2017 (hasil optimasi)	0.978827

Berdasarkan Tabel 6.11 menyebutkan bahwa nilia *jain fairness index* jadwal hasil generating otomatis bulan Desember 2017 lebih tinggi dari jadwal manual November 2017. Sedangkan hasil optimasi pada jadwal bulan Desember 2017 memiliki nilai JFI sebesar 0.978827, meningkat sebesar 20% dari nilai JFI semula sebelum jadwal dioptmasikan. Hal tersebut menyatakan tingkat keadilan yang sangat tinggi dikarenakan nilai mendekati satu.

Dari hasil optimasi penjadwalan unit IGD dan Rawat Jalan sesuai dengan lampiran D, pada Tabel 6.12 dibawah ini

dilakukan perhitungan komposisi shift setiap staf yang dijadwalkan. Berdasarkan Tabel 6.12 dapat dilihat bahwa optimasi penjadwalan dapat memberikan nilai *fairness* kepada pembagian hari libur sesuai dengan pembobotan akan tetapi kurang bagus dalam pemerataan jam kerja.

Tabel 6.12 Jumlah shift setiap staf unit IGD dan Rawat jalan

Id Staf	Tipe Shift									Total Jam Kerja
	P	S	MS	MD	L	CT	M	LM	Total	
301	4	11	3	0	6	0	5	2	31	154
302	3	12	2	2	7	0	4	1	31	154
303	5	12	2	1	5	0	4	2	31	168
304	4	13	1	1	5	0	4	3	31	175
305	4	13	2	1	5	0	4	2	31	168
306	5	12	2	1	5	0	4	2	31	168
307	4	13	1	2	4	0	5	2	31	182
308	3	13	2	1	5	0	5	2	31	168

Pada Tabel 6.13 akan dijabarkan pembagagian hari libur antar staf pada unit. Pembagian hari libur akan dihitung berdasarkan jumlah hari libur pada hari sabtu, jumlah libur pada hari minggu dan jumlah libur pada hari kerja atau *weekdays*.

Tabel 6.13 Jumlah hari libur setiap staf unit IGD dan Rawat Jalan

Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
301	1	0	1	1	1	1	1	6
302	0	1	0	0	2	2	2	7
303	1	2	1	0	0	0	1	5
304	2	0	2	1	0	0	0	5
305	0	0	0	2	2	0	1	5
306	1	0	0	0	0	2	2	5

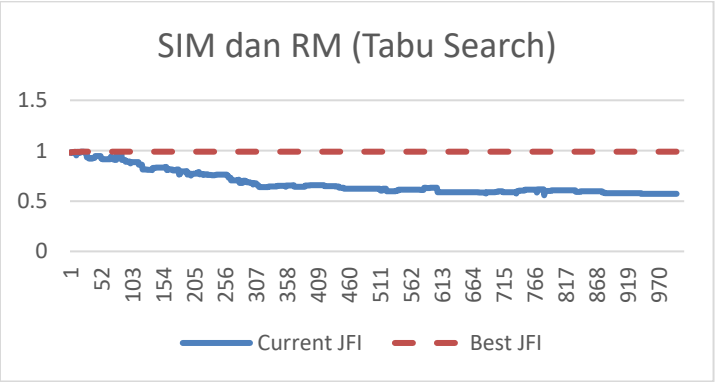
Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
307	1	2	0	0	0	0	1	4
308	1	0	1	1	1	1	0	5

Dimana M merupakan hari mingu, Sb merupakan hari sabtu, Sn merupakan hari senin, Sl merupakan hari selasa, R merupakan hari rabo, K merupakan hari kamis, dan J merupakan hari Jum'at.

6.5. Hasil Implementasi Penjadwalan Staf Unit SIM dan RM

Hasil implementasi penjadwalan staf unit SIM dan RM dengan menggunakan *tabu search based hyper heuristics* dapat dilihat pada gambar lampiran D. Hasil optimasi penjadwalan dipilih berdasakan nilai paling optimum yang dihasilkan dari kode program yang dibangun.

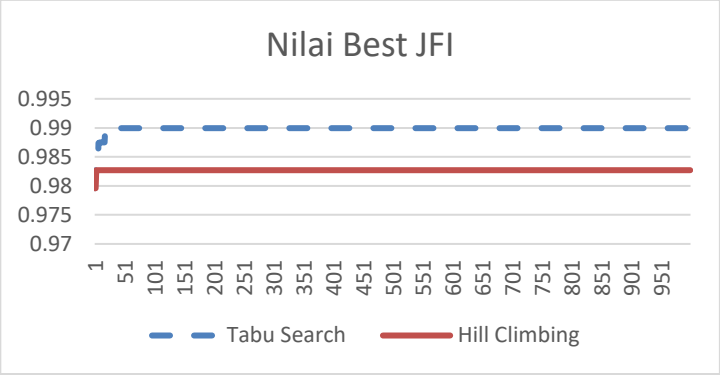
Langkah pertama yang dilakukan penulis adalah melakukan optimasi dengan menggunakan algoritma *tabu search* pada jadwal unit SIM dan RM dengan parameter masukan iterasi sebanyak seribu.



Gambar 6.10 Hasil 1000 iterasi algoritma tabu search pada unit SIM dan RM

Berdasarkan Gambar 6.10 menyebutkan semakin tinggi tingkat itersi pada algoritma *tabu search* menghasilkan *cost* JFI yang semakin turun secara eksponential. Nilai JFI naik mulai iterasi pertama sampai iterasi ke 100 sampai 200. Hal tersebut dapat diketahui bahwa performa algoritma *tabu search* dengan data jadwal unit SIM dan RM memiliki kemampuan cukup cepat untuk menemukan solusi terbaik.

Langkah kedua adalah membandingkan *best* JFI yang didapatkan dari algoritma *tabu search* dan algoritma *hill climbing*. Hasil *best* JFI didapatkan dengan parameter iterasi sebanyak seribu dengan satu kali eksekusi program.



Gambar 6.11 Perbandingan nilai JFI terbaik unit SIM dan RM

Berdasarkan Gambar 6.11 algoritma *tabu search* menghasilkan best JFI lebih tinggi dari algoritma *hill climbing*. Algoritma *tabu search* dan *hill climbing* sama-sama memiliki kecepatan untuk mendapatkan solusi lebih cepat yaitu pada iterasi ke 50 sampai seratus.

Berdasarkan hasil percobaan pada Gambar 6.10 dan Gambar 6.11 maka dapat digunakan sebagai acuan berapa kali iterasi yang baik untuk melakukan optimasi. Sehingaa penulis

berasumsikan untuk melakukan percobaan iterasi sebanyak seratus kali iterasi dan dua ratus iterasi dengan percobaan optimasi sebanyak dua puluh satu kali. Tabel 6.14 dibawah ini merupakan hasil uji coba optimasi dengan menggunakan algoritma *tabu search* dan *hill climbing*.

Tabel 6.14 Hasil uji coba optimasi unit SIM dan RM

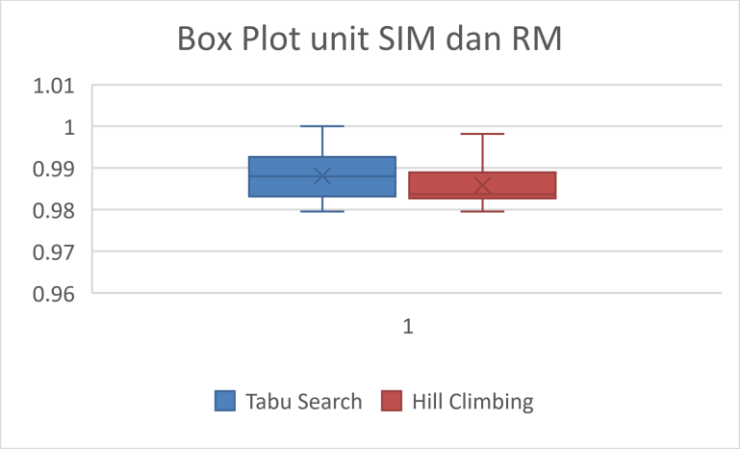
NO	Algoritma	Jumlah Iterasi	Nilai	
1	**Tabu Search**	**100**	Max	1
			Min	0.979592
			Avg	0.986782
		200	Max	0.998223
			Min	0.982704
			Avg	0.989366
2	**Hill Climbing**	**100**	Max	0.998223
			Min	0.979592
			Avg	0.986564
		200	Max	0.998223
			Min	0.979592
			Avg	0.985206

Berdasarkan infromasi pada Tabel 6.14, nilai *jains fairness index* paling tinggi didapatkan dengan menggunakan algoritma *tabu search* dengan jumlah iterasi sebanyak seratus kali, yaitu sebesar 1. Dan hasil *jains fairness index* terkecil didapatkan pada optimasi dengan menggunakan algoritma *tabu search* dengan iterasi seratus kali dan dengan menggunakan algoritma *hill climbing* dengan iterasi sebanyak dua ratu dan seratus kali yaitu sebesar 0.979592. Rincian hasil secara lengkap terdapat pada lampiran C.

Untuk mengetahui persebaran hasil optimasi kedua algoritma berikut ini merupakan gambar Box plot pada Gambar 6.12

berdasarkan box plot, kedua algoritma memiliki hasil yang sama yaitu memiliki rata rata pada angka 0.9.

Nilai JFI maksimal didapatkan dari algoritma *tabu search* dan algoritma *hill climbing* dengan iterasi sebanyak seratus kali dan nilai JFI terendah didapatkan dengan algoritma *tabu search* dengan jumlah iterasi sebanyak dua ratus kali dan algoritma *hill climbing* dengan iterasi sebanyak seratu kali dan dua ratus kali.



Gambar 6.12 Box plot unit SIM dan RM

Perbandingan nilai JFI atau *jain fairness index* digunakan untuk mengetahui nilai JFI yang paling maksimal. Perbandingan JFI dilihat dari jadwal manual bulan November 2017, jadwal otomatis Desember 2017, dan jadwal Desember 2017 hasil optimasi.

Tabel 6.15 Perbandingan nilai JFI Unit SIM dan RM

Unit SIM dan RM		
No	**Jadwal pada Bulan ke-**	**JFI**
1	Jadwal Manual November 2017	0.746714456
2	Jadwal Otomatis Desember 2017	0.975918367
3	Jadwal Desember 2017 (hasil optimasi)	1

Berdasarkan Tabel 6.15 menyebutkan bahwa nilia *jain fairness index* jadwal hasil generating otomatis bulan Desember 2017 lebih tinggi dari jadwal manual November 2017. Sedangkan hasil optimasi pada jadwal bulan Desember 2017 memiliki nilai JFI sebesar 1, meningkat sebesar 2% dari nilai JFI semula sebelum jadwal dioptmasikan. Sehingaa optimasi jadwal pada unit SIM dan RS menghasilkan nilai keadilan yang sanagt sempurna dikarenakan nilai JFI sema dengan satu.

Dari hasil optimasi penjadwalan unit SIM dan RM sesuai dengan lampiran D, pada Tabel 6.16 dibawah ini dilakukan perhitungan komposisi shift setiap staf yang dijadwalkan. Berdasarkan Tabel 6.16 dapat dilihat bahwa optimasi penjadwalan dapat memberikan nilai *fairness* kepada pembagian hari libur sesuai dengan pembobotan akan tetapi kurang bagus dalam pemerataan jam kerja.

Tabel 6.16 Jumlah shift setiap staf unit SIM dan RM

Id Staf	Tipe Shift								Total Jam Kerja
	P	S	MS	L	CT	M	LM	Total	
401	10	5	5	5	0	4	2	31	182
402	9	6	4	6	0	4	2	31	175
403	7	5	4	7	0	5	3	31	168
404	8	4	4	6	0	6	3	31	175
405	7	4	5	5	0	7	3	31	182
406	9	7	0	6	0	6	3	31	175

Pada Tabel 6.17 akan dijabarkan pembagagian hari libur antar staf pada unit. Pembagian hari libur akan dihitung berdasarkan jumlah hari libur pada hari sabtu, jumlah libur pada hari minggu dan jumlah libur pada hari kerja atau *weekdays*.

Tabel 6.17 Jumlah hari libur setiap staf unit SIM dan RM

Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
401	1	1	0	0	1	1	1	5
402	1	0	1	2	0	1	1	6
403	0	2	1	1	1	2	0	7
404	0	2	1	0	1	1	1	6
405	1	1	1	0	1	0	1	5
406	1	1	1	1	1	0	1	6

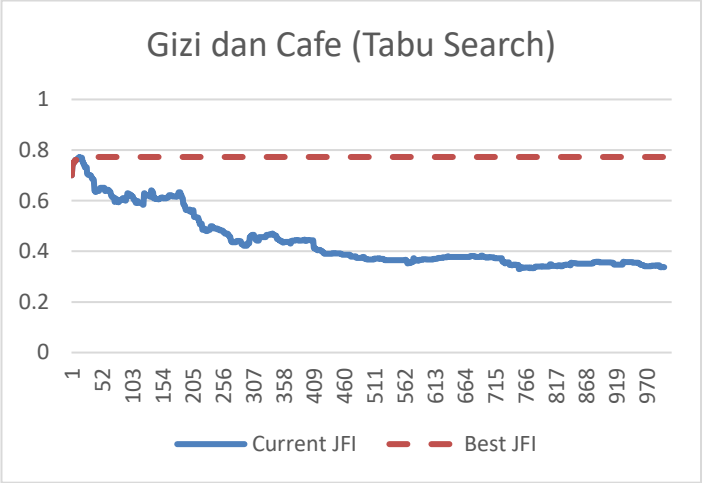
Dimana M merupakan hari mingu, Sb merupakan hari sabtu, Sn merupakan hari senin, Sl merupakan hari selasa, R merupakan hari rabo, K merupakan hari kamis, dan J merupakan hari Jum'at.

6.6. Hasil Implementasi Penjadwlan Staf Unit Gizi dan Café

Hasil implementasi penjadwalan staf unit Gizi dan Cafe dengan menggunakan *tabu search based hyper heuristics* dapat dilihat pada gambar lampiran D. Hasil optimasi penjadwalan dipilih berdasakan nilai paling optimum yang dihasilakan dari kode program yang dibangun.

Langkah pertama yang dilakukan penulis adalah melakukan optimasi dengan menggunakan algoritma *tabu search* pada jadwal unit Gizi dan Cafe dengan parameter masukan iterasi sebanyak seribu.

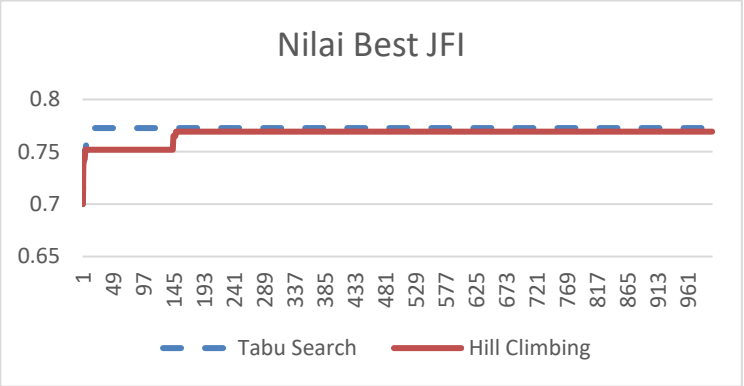
Berdasarkan Gambar 6.13 di bawah menyebutkan semakin tinggi tingkat itersi pada algoritma *tabu search* menghasilkan *cost* JFI yang semakin turun secara eksponential. Nilai JFI naik mulai iterasi pertama sampai iterasi ke lima puluh. Hal tersebut dapat diketahui bahwa performa algoritma *tabu search* dengan data jadwal unit Gizi dan Cafe memiliki kemampuan cukup cepat untuk menemukan solusi terbaik.



Gambar 6.13 Hasil 1000 iterasi algoritma tabu search pada unit Gizi dan Café

Langkah kedua adalah membandingkan *best* JFI yang didapatkan dari algoritma *tabu search* dan algoritma *hill climbing*. Hasil *best* JFI didapatkan dengan parameter iterasi sebanyak seribu dengan satu kali eksekusi program.

Berdasarkan Gambar 6.14 algoritma *tabu search* menghasilkan best JFI lebih tinggi dari algoritma *hill climbing*. Algoritma *tabu search* memiliki kecepatan mendapatkan solusi terbaik lebih cepat dari pada algotitma *hill climbing*. Algoritma tabu search menemukan solusi terbaik pada iterasi ke 50, sedangkan algoritma *hill climbing* memiliki kecepatan untuk mendapatkan solusi terbaik pada iterai ke 190.



Gambar 6.14 Perbandingan nilai JFI terbaik unit Gizi dan café

Berdasarkan hasil percobaan pada gambar Gambar 6.13 dan Gambar 6.14 maka dapat digunakan sebagai acuan berapa kali iterasi yang baik untuk melakukan optimasi. Sehingaa penulis berasumsikan untuk melakukan percobaan iterasi sebanyak seratus kali iterasi dan dua ratus iterasi dengan percobaan optimasi sebanyak dua puluh satu kali. Tabel 6.18 dibawah ini merupakan hasil uji coba optimasi dengan menggunakan algoritma *tabu search* dan *hill climbing*.

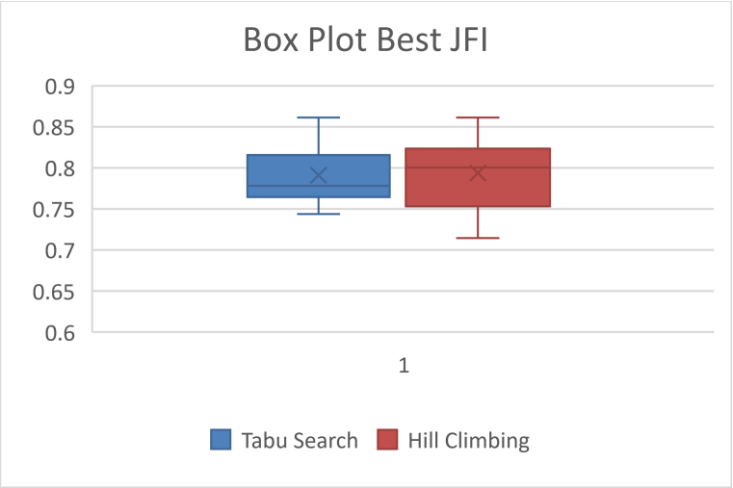
Tabel 6.18 Hasil uji coba optimasi unit Gizi dan Cafe

NO	Algoritma	Jumlah Iterasi	Nilai	
1	Tabu Search	100	Max	0.861071
			Min	0.743789
			Avg	0.788966
		200	Max	0.860951
			Min	0.743789
			Avg	0.792502
2	Hill Climbing	100	Max	0.857769
			Min	0.714176
			Avg	0.786249
		200	Max	0.861071

NO	Algoritma	Jumlah Iterasi	Nilai	
			Min	0.72945
			Avg	0.800686

Berdasarkan infromasi pada Tabel 6.18, nilai *jains fairness index* paling tinggi didapatkan dengan menggunakan algoritma *tabu search* dengan jumlah iterasi sebanyak seratus kali, yaitu sebesar 0.861071. Dan hasil *jains fairness index* terkecil didapatkan pada optimasi dengan menggunakan algoritma *tabu search* dengan iterasi dua ratus kali yaitu sebesar 0.714176. Rincian hasil secara lengkap terdapat pada lampiran C.

Untuk mengetahui persebaran hasil optimasi kedua algoritma berikut ini merupakan gambar Box plot pada Gambar 6.15., kedua algoritma memiliki hasil yang sama yaitu memiliki rata rata pada angka 0.9.



Gambar 6.15 Box plot Best JFI unit Gizi dan Café

Nilai JFI maksimal didapatkan dari algoritma *tabu search* dan algoritma *hill climbing* dengan iterasi sebanyak seratus kali dan

nilai JFI terendah didapatkan dengan algoritma *tabu search* dengan jumlah iterasi sebanyak dua ratus kali.

Perbandingan nilai JFI atau *jain fairness index* digunakan untuk mengetahui nilai JFI yang paling maksimal. Perbandingan JFI dilihat dari jadwal manual bulan November 2017, jadwal otomatis Desember 2017, dan jadwal Desember 2017 hasil optimasi.

Tabel 6.19 Perbandingan nilai JFI Unit Gizi dan Cafe

Unit Gizi		
No	**Jadwal pada Bulan ke-**	**JFI**
1	Jadwal Manual November 2017	0.901225667
2	Jadwal Otomatis Desember 2017	0.700040177
3	Jadwal Desember 2017 (hasil optimasi)	0.861071

Berdasarkan Tabel 6.19 menyebutkan bahwa nilia *jain fairness index* jadwal manual November 2017 lebih tinggi dari jadwal hasil generating otomatis bulan Desember 2017. Sedangkan hasil optimasi pada jadwal bulan Desember 2017 memiliki nilai JFI sebesar 0.861071, meningkat sebesar 23% dari nilai JFI semula sebelum jadwal dioptmasikan. Hal tersebut menyatakan tingkat keadilan yang sangat tinggi dikarenakan nilai mendekati satu.

Dari hasil optimasi penjadwalan unit Gizi dan cafe sesuai dengan lampiran D, pada Tabel 6.20 dibawah ini dilakukan perhitungan komposisi shift setiap staf yang dijadwalkan. Berdasarkan Tabel 6.20 dapat dilihat bahwa optimasi penjadwalan dapat memberikan nilai *fairness* kepada pembagian hari libur sesuai dengan pembobotan akan tetapi kurang bagus dalam pemerataan jam kerja.

Tabel 6.20 Jumlah shift setiap staf unit Gizi dan cafe

Id Staf	Tipe Shift												Total Jam Kerja
	P	P1	P2	PS	S	MD	MS	L	CT	M	LM	Total	
501	27	0	0	0	1	0	0	3	0	0	0	31	196
502	5	0	5	0	9	6	0	6	0	0	0	31	98
503	6	0	6	1	9	7	0	2	0	0	0	31	105
504	6	0	6	0	10	6	0	3	0	0	0	31	112
505	7	0	6	0	9	6	0	3	0	0	0	31	112
506	0	0	12	0	12	0	0	7	0	0	0	31	84
507	9	0	0	7	9	0	0	6	0	0	0	31	126
508	11	0	1	5	9	0	0	5	0	0	0	31	140
509	9	0	1	5	8	0	0	8	0	0	0	31	119
510	9	0	1	0	10	8	0	3	0	0	0	31	133
511	12	0	1	0	6	0	0	3	0	7	2	31	189
512	13	0	1	0	7	0	0	1	0	6	3	31	203
513	10	0	0	0	8	1	0	3	0	6	3	31	189
514	13	0	0	0	6	0	0	3	0	6	3	31	196
515	7	0	0	0	5	4	0	5	0	6	4	31	154
516	10	4	1	0	13	0	0	3	0	0	0	31	161
517	12	5	0	0	10	0	0	4	0	0	0	31	154
518	12	5	0	0	10	0	0	4	0	0	0	31	154
519	10	4	0	0	12	0	0	4	0	0	1	31	161

Pada Tabel 6.21 akan dijabarkan pembagagian hari libur antar staf pada unit. Pembagian hari libur akan dihitung berdasarkan jumlah hari libur pada hari sabtu, jumlah libur pada hari minggu dan jumlah libur pada hari kerja atau *weekdays*.

Tabel 6.21 Jumlah hari libur setiap staf unit Gizi dan Cafe

Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
501	3	0	0	0	0	0	0	3
502	1	1	0	1	1	1	1	6
503	1	0	0	0	0	0	1	2
504	1	1	0	0	1	0	0	3
505	0	1	0	1	0	0	1	3
506	0	2	0	0	2	2	1	7
507	1	1	1	1	0	1	1	6
508	1	1	0	1	2	0	0	5
509	1	1	2	1	0	2	1	8
510	0	0	0	0	0	3	0	3
511	2	0	1	0	0	0	0	3
512	0	0	0	1	0	0	0	1
513	1	0	1	0	0	0	1	3
514	1	1	0	0	1	0	0	3
515	0	1	1	1	0	1	1	5
516	2	0	1	0	0	0	0	3
517	2	0	2	0	0	0	0	4
518	2	0	2	0	0	0	0	4
519	2	0	2	0	0	0	0	4

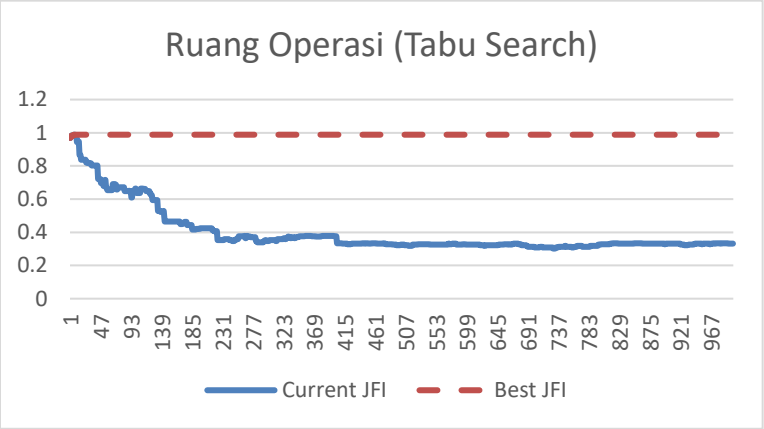
Dimana M merupakan hari mingu, Sb merupakan hari sabtu, Sn merupakan hari senin, Sl merupakan hari selasa, R merupakan

hari rabo, K merupakan hari kamis, dan J merupakan hari Jum'at.

6.7. Hasil Implementasi Penjadwalan Staf Ruang Operasi

Hasil implementasi penjadwalan staf Ruang Operasi dengan menggunakan *tabu search based hyper heuristics* dapat dilihat pada gambar lampiran D. Hasil optimasi penjadwalan dipilih berdasakan nilai paling optimum yang dihasilkan dari kode program yang dibangun. Sifat *hyper heuristics* yang kemunculanya tidak pasti maka penulis melakukan eksplorasi pada dua algoritma yaitu *hill climbing* dan *tabu search* untuk menentukan hasil yang paling optimal.

Langkah pertama yang dilakukan penulis adalah melakukan optimasi dengan menggunakan algoritma *tabu search* pada jadwal unit farmasi dengan parameter masukan iterasi sebanyak seribu.

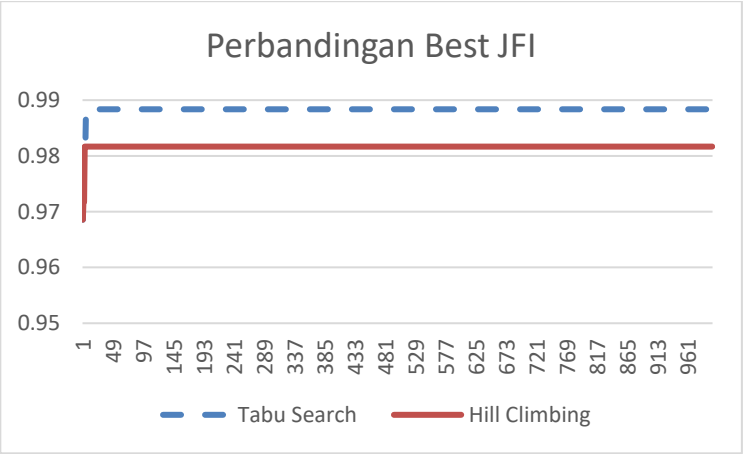


Gambar 6.16 Hasil 1000 iterasi algoritma tabu search pada Ruang Operasi

Berdasarkan Gambar 6.16 di atas menyebutkan semakin tinggi tingkat itersi pada algoritma *tabu search* menghasilkan *cost* JFI

yang semakin turun secara eksponential. Nilai JFI naik mulai iterasi pertama sampai iterasi ke lima puluh. Hal tersebut dapat diketahui bahwa performa algoritma *tabu search* dengan data jadwal Ruang Operasi memiliki kemampuan cukup cepat untuk menemukan solusi terbaik.

Langkah kedua adalah membandingkan *best* JFI yang didapatkan dari algoritma *tabu search* dan algoritma *hill climbing*. Hasil *best* JFI didapatkan dengan parameter iterasi sebanyak seribu dengan satu kali eksekusi program.



Gambar 6.17 Perbandingan nilai best JFI Ruang Operasi

Berdasarkan Gambar 6.17 algoritma *tabu search* menghasilkan best JFI lebih tinggi dari algoritma *hill climbing*. Algoritma *tabu search* memiliki kecepatan mendapatkan solusi terbaik lebih cepat dari pada algotitma *hill climbing*. Algoritma tabu search dan hill climbing memiliki kecepatan yang sama dalam menemukan solusi terbaik. Solusi terbaik ditemukan pada saat program memasuki iterasi ke lima puluh. cepatan untuk mendapatkan solusi terbaik pada iterai ke 190.

Berdasarkan hasil percobaan pada Gambar 6.16 dan Gambar 6.17 maka dapat digunakan sebagai acuan berapa kali iterasi yang baik untuk melakukan optimasi. Sehingaa penulis berasumsikan untuk melakukan percobaan iterasi sebanyak lima puluh kali iterasi dan seratus kali iterasi dengan percobaan optimasi sebanyak dua puluh satu kali. Tabel 6.22 dibawah ini merupakan hasil uji coba optimasi dengan menggunakan algoritma *tabu search* dan *hill climbing*.

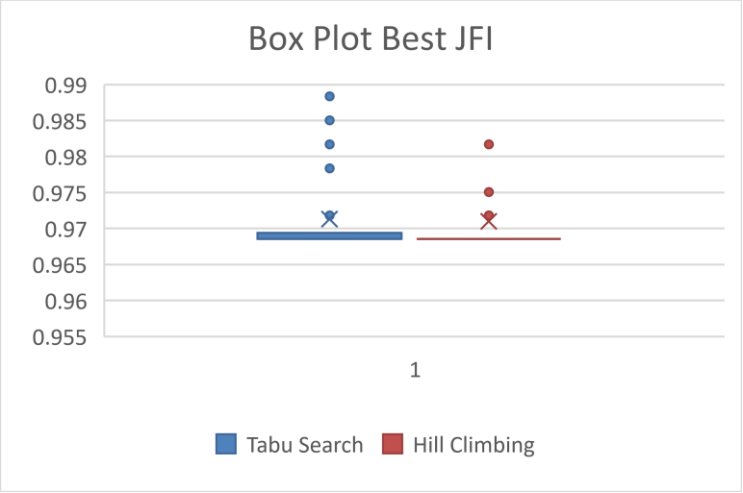
Tabel 6.22 Hasil uji coba optimasi Ruang Operasi

NO	Algoritma	Jumlah Iterasi	Nilai	
1	**Tabu Search**	**100**	Max	0.988359
			Min	0.968559
			Avg	0.971688
		200	Max	0.981669
			Min	0.968559
			Avg	0.970895
2	**Hill Climbing**	**100**	Max	0.981669
			Min	0.968559
			Avg	0.970896
		200	Max	0.981669
			Min	0.968559
			Avg	0.971056

Berdasarkan infromasi pada Tabel 6.22, nilai *jains fairness index* paling tinggi didapatkan dengan menggunakan algoritma *tabu search* dengan jumlah iterasi sebanyak seratus kali, yaitu sebesar 0.988359. Dan hasil *jains fairness index* terkecil didapatkan pada optimasi dengan menggunakan algoritma *hill climbing* dengan iterasi seratus kali yaitu sebesar 0.968559. Rincian hasil secara lengkap terdapat pada lampiran C.

Untuk mengetahui persebaran hasil optimasi kedua algoritma berikut ini merupakan gambar Box plot pada Gambar 6.18.

berdasarkan box plot, kedua algoritma memiliki hasil yang sama yaitu memiliki rata rata pada angka 0.97.



Gambar 6.18 Box plot Ruang Operasi

Perbandingan nilai JFI atau *jain fairness index* digunakan untuk mengetahui nilai JFI yang paling maksimal. Perbandingan JFI dilihat dari jadwal manual bulan November 2017, jadwal otomatis Desember 2017, dan jadwal Desember 2017 hasil optimasi.

Tabel 6.23 Perbandingan nilai JFI Ruang Operasi

Kamar Operasi		
No	**Jadwal pada Bulan ke-**	**JFI**
1	Jadwal Manual November 2017	0.89031339
2	Jadwal Otomatis Desember 2017	0.968558709
3	Jadwal Desember 2017 (hasil optimasi)	0.988359

Berdasarkan Tabel 6.23 menyebutkan bahwa nilia *jain fairness index* jadwal hasil generating otomatis bulan Desember 2017 lebih tinggi dari jadwal manual November 2017. Sedangkan

hasil optimasi pada jadwal bulan Desember 2017 memiliki nilai JFI sebesar 0.988359, meningkat sebesar 47% dari nilai JFI semula sebelum jadwal dioptmasikan. Hal tersebut menyatakan tingkat keadilan yang sangat tinggi dikarenakan nilai mendekati satu.

Dari hasil optimasi penjadwalan Ruang Operasi sesuai dengan lampiran D, pada Tabel 6.24 dibawah ini dilakukan perhitungan komposisi shift setiap staf yang dijadwalkan. Berdasarkan Tabel 6.24 dapat dilihat bahwa optimasi penjadwalan dapat memberikan nilai *fairness* kepada pembagian hari libur sesuai dengan pembobotan akan tetapi kurang bagus dalam pemerataan jam kerja.

Tabel 6.24 Jumlah shift setiap staf Ruang Operasi

Id Staf	Tipe Shift					Total Jam Kerja
	P	S	L	CT	Total	
601	17	10	4	0	31	189
602	17	10	4	0	31	189
603	15	11	5	0	31	182
604	18	9	4	0	31	189
605	20	7	4	0	31	189
606	19	7	5	0	31	182

Pada Tabel 6.25 akan dijabarkan pembagagian hari libur antar staf pada unit. Pembagian hari libur akan dihitung berdasarkan jumlah hari libur pada hari sabtu, jumlah libur pada hari minggu dan jumlah libur pada hari kerja atau *weekdays*.

Tabel 6.25 Jumlah hari libur setiap staf Ruang Operasi

Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
601	2	0	0	1	1	0	0	4
602	2	0	0	0	1	1	0	4
603	1	1	1	0	0	1	1	5

Id Staf	Libur							Total Libur
	M	Sb	Sn	Sl	R	K	J	
604	1	1	0	1	0	0	1	4
605	2	1	0	0	1	0	0	4
606	2	0	1	0	0	1	1	5

Dimana M merupakan hari mingu, Sb merupakan hari sabtu, Sn merupakan hari senin, Sl merupakan hari selasa, R merupakan hari rabo, K merupakan hari kamis, dan J merupakan hari Jum’at.

BAB VII
KESIMPULAN DAN SARAN

Pada bagian ini akan dijelaskan hasil implementasi dari rancangan model yang telah dibuat sebelumnya. Bagian ini akan menjelaskan perbandingan hasil antara menggunakan algoritma *tabu search hyper-heuristics* dengan jadwal yang sudah digunakan Rumah Sakit Ibu dan Anak Kendangsari saat ini.

7.1. Kesimpulan

Kesimpulan yang dapat diambil dari Tugas Akhir ini adalah pembuatan jadwal yang optimal dapat diselesikan dengan menggunakan algoritma *tabu search based hyper heuristics*.

- Semua *hard constraint* yang dimodelkan pada setiap unit baik jadwal hasil *generating feasible schedule* maupun jadwal hasil optimasi dapat terpenuhi.
- Hasil nilai *fairness* antara jadwal periode November 2017 dari rumah sakit dengan jadwal desember 2017 hasil optimasi semua unit dapat ditingkatkan kecuali unit gizi.
- Nilai *fairness* pada Penelitian ini hanya menggunakan faktor pembagian hari libur. Pada dasarnya tingkat *fairness* tidak hanya ditentukan faktor penentuan hari libur. Banyak faktor lain yang dapat digunakan untuk mengukur tingkat *fairness* dalam pembentukan jadwal. Sehingg perlu eksplorasi lanjut terhadap presferensi yang dimiliki oleh masing-masing staf untuk meningkatkan tingkat keadilan penjadwalan.
- Dibandingkan dengan hasil optimasi, jadwal manual unit gizi dan café memiliki nilai JFI lebih tingi yaitu 0.901225667, dibandingkan dengan hasil optimasi yaitu hanya sebesar 0.861071, akan tetapi nilai JFII hasil optimasi lebih besar dari pada jadwal hasil *generating* yaitu hanya memiliki nilai JFI sebesar 0.700040177.

- Nilai *jain fairness index* antara hasil *generating* otomatis dengan hasil optimasi mengalami peningkatan. Pada unit Farmasi meningkat sebesar 47%, unit Nicu & Ruang Bayi meningkat sebesar 48%, unit IGD meningkat sebesar 20%, unit SIM & RM meningkat sebesar 2%, unit Gizi & Café meningkat sebesar 23% dan Ruang Operasi meningkat sebesar 2%.

7.2. Saran

Berdasarkan hasil dan kesimpulan diatas, saran yang dapat diberikan untuk penelitian Tugas Akhir ini adalah sebagai berikut:

- Penelitian ini hanya menggunakan metode *tabu serach hyper heuristics* dimana algoritma *tabu search* masih memiliki banyak kelemahan diantaranya proses iteriasi yang dibutuhkan sangat banyak untuk mencari solusi selain itu banyaknya parameter yang harus ditentukan peneliti juga menjadi kelemahan algoritma *tabu search*. Untuk mencari nilai yang lebih optimal dengan cara yang lebih efisien maka Penelitian selanjutnya dapat mencoba metode atau algoritma yang lebih *advance* dari *tabu search*.
- Pada Penelitian ini hanya menggunakan *hard constraint* per-hari. Sehinga pada Penelitian selanjutnya hendaknya menambahkan beberapa batasan yang harus dipenuhi dalam melakukan optimasi seperti batasan masing masing skil staf yang dijadwalkan, batasan setiap hari, batasan setiap bulan atau batasan setiap tahun.
- Pada Penelitian ini nilai *fairness* diambil dari nilai keadilan nilai hari libur. Pada Penelitian selanjutnya hendaknya ukuran *fairness* dalam penjadwalan perawat bisa diambil dari faktor lain seperti pembagian jam kerja antara staf.
- Pada Penelitian ini hanya menggunakan *low level heuristics* sebanyak dua yaitu *move* dan *swap*. Untuk

menemukan variansi solusi yang optimal maka perlu adanya *low level heuristics* lebih agar proses optimasi dapat memilih lebih banyak *heuristics* sehingga hasil optimasi lebih dapat dioptimalkan.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

[1] B. Cheang, A. Lim, and B. Rodrigues, "Nurse Rostering Problems : A Bibliographic Survey Nurse Rostering Problems : A Bibliographic Survey," pp. 447–460, 2003.

[2] G. Kendall and N. M. Hussin, "A Tabu Search Hyper-heuristic Approach to the Examination Timetabling Problem at the MARA University of Technology," pp. 270–293, 2005.

[3] F. Glover and M. Laguna, "Principles of Tabu Search," 2005.

[4] E. Burke, "SCHEDULING NURSES USING A TABU-SEARCH HYPERHEURISTIC," pp. 1–22, 2003.

[5] E. K. Burke, G. Kendall, and E. Soubeiga, "A Tabu-Search Hyperheuristic for Timetabling," pp. 451–470, 2004.

[6] T. Curtois, "(2016) A tensor based hyper-heuristic for nurse A Tensor Based Hyper-heuristic for Nurse Rostering," pp. 185–199, 2016.

[7] J. Z. & X. Z. Xiang Zhong, "A two-stage heuristic algorithm for the nurse scheduling problem with fairness objective on weekend workload under different shift designs," 2017.

[8] M. A. Amir, *OPTIMASI PENJADWALAN PERAWAT MENGGUNAKAN GABUNGAN INTEGER LINEAR PROGRAMMING DAN VARIABLE NEIGHBORHOOD SEARCH. STUDI KASUS INSTALASI GAWAT DARURAT RUMAH SAKIT IBNU SINA MAKASSAR*. 2017.

[9] T. World and H. Report, "for health," 2006.

[10] *UU No.40 tahun 2009*. 2009.

[11] "Undang Undang tentang Rumah Sakit." [Online]. Available: http://www.depkes.go.id/resources/download/peraturan/UU No. 44 Th 2009 ttg Rumah Sakit.PDF. [Accessed:

17-Sep-2017].
[12] "Profil RSIA Merr Kendangsari." [Online]. Available: http://merr.kendangsari.com/tentang-kami/. [Accessed: 17-Sep-2017].
[13] "Layanan dan fasilitas Unggulan RSIA." [Online]. Available: http://merr.kendangsari.com/layanan-unggulan/. [Accessed: 17-Sep-2017].
[14] Dr. YAKOOB. C, "MANAGEMENT SCIENCE," 2016.
[15] C. West Churchman, Russell L. Ackoff & E. L. Arnoff, *Introduction to Operations Research*. New York, 1957.
[16] *Permenkes No.56*, vol. 2008. 2014.
[17] H. Professionals, "Classifying health workers : Mapping occupations to the international standard classification," pp. 1–14, 2008.
[18] P. Smet and S. Martin, "Investigation of fairness measures for nurse rostering."
[19] D. Ouelhadj, S. Martin, P. Smet, and E. Ozcan, "Fairness in Nurse Rostering," no. 1990, pp. 1–19.
[20] D.-M. Jain, Rajendra K. W, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared ComputeredSystem," 1984.
[21] J. Deng and Y. S. Han, "Fairness Index Based on Variational Distance."
[22] A. Muklason and A. J. Parkes, "Initial Results on Fairness in Examination Timetabling," no. August 2010, pp. 27–29, 2013.
[23] P. R. Burke, E. Hart, G. Kendall, J. Newall, "Schulenburg, Hyper-heuristics: An emerging direction in modern search technology, Handbook of Metaheuristics," Kluwer, 2003, p. 457–474.
[24] A. Muklason, *PhD Thesis The University of Nottingham Hyper-heuristics and Fairness in Examination Timetabling Problems*. 2017.
[25] G. Kendall and N. M. Hussin, "AN INVESTIGATION OF A TABU SEARCH BASED HYPER-HEURISTIC FOR EXAMINATION TIMETABLING Hyper-

heuristics," no. 1986.

[26] P. Russell, S; Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, New Jersey, 2003.

[27] H. Wang, D. Wang, and S. Yang, "A Memetic Algorithm with Adaptive Hill Climbing Strategy for Dynamic Optimization Problems," vol. 13, 2009.

[28] "Tabu Search." [Online]. Available: http://elib.unikom.ac.id/files/disk1/307/jbptunikompp-gdl-restinovri-15327-3-13_babii.pdf. [Accessed: 27-Sep-2017].

[29] D. Pham, D.T. and Karaboga, "Intelligent Optimisation Techniques – Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks," 2000.

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Penulis Tugas Akhir ini lahir di Lamongan Jawa Timur pada tanggan 30 Juli 1996 dengan nama lengkap Zuli Maulidati. Penulis dengan sapaan akrab Zuli ini merupakan anak kedua dari tiga bersaudara dari Ayah yang bernama Muhammad Dalil dan Ibu yang bernama Mufidah. Belum genap berusia satu tahun, Penulis dan keluarga pindah tempat tinggal di kabupaten Gresik karena Pindah tugas sang Ayah yakni tepatnya di Desa Wotan Kecamatan Panceng Gresik yang sampai saat ini. Pendidikan tingkat Sekolah Dasar dihabiskan penulis di MI Muhammadiyah 04 Wotan Gresik dan lulus pada tahun 2008. Di jenjang menengah pertama penulis melanjutkan studi di MTs Muhammadiyah 09 Wotan Panceng Gresik dan mendapatkan ijazah kelulusan pada tahun 2011. Pada tahun itulah penulis memutuskan untuk memasuki pesantren dalam menikmati masa masa Sekolah Menengah Atas yaitu tepatnya di MA Al Ishlah Sendangaung Paciran Lamongan dan pondok pesantren Al Ishlah Sendangagung Paciran Lamongan. Setelah menamatkan pendidikan Aliyah pada tahun 2014 penulis melanjutkan menuntut ilmu di Institut Teknologi Sepuluh Nopember Departemen Sistem Informasi Fakultas Teknologi Informasi hingga sekarang. Selama perkuliahan, penulis aktif sebagai panitia kegiatan baik tingkat jurusan maupun fakultas dengan menjadi panitia Information System Expo (ISE), FTIf Journey dan Pelatihan Jurnalistik Tingkat Menengah FTIf 2016. Penulis juga aktif berorganisasi di Badan Eksekutif Mahasiswa FTIf, CSSMoRA ITS, Kajian Sistem Infromasi, dan Trainer Keilmiahan ITS Integrator.

Di departemen Sistem Informasi, penulis mengambil bidang minat Rekayasa Data dan Integelgnsia Bisnis. Untuk mengetahui mengetahui lebih jelas terkait dengan Penelitian ini, penulis dapat dihubunga melalui email zmaulidati@@gmail.com.

LAMPIRAN A: Hasil Wawancara

Tabel A.1 Interview Protocol

Interview Protocol	
Human Resource Departemen view	
Informasi Interview	
Interviewer	Zuli Maulidati
Narasumber	Ibu Silvy
Hari tanggal	Senin, 28 Oktober 2017
Pukul	10.30-12.30
Lokasi	RSIA Merr Kendnagsari
Informasi Narasumber	
Nama	Bu Sylvy Medtasya Dzykrzyanka, S. Farm, M. Farm. Klin, APT, MARS
Jabatan	Kepala Bagian
Divisi	Sumber Daya Manusia (SDM) & Hukum
Instansi	RSIA Merr Kendangsari
Lama Bekerja	3.5 tahun
Penjelasan Interview	Interview ini bertujuan untuk salah satu sumber data untuk tugas akhir dengan judul "Optimasi Penjadwalan Staff rumah sakit Dengan Menggunakan Algoritma *Tabu Search Based Hyper-Heuristics* (Studi Kasus: Rumah Sakit Ibu Dan Anak Kendangsari)" yang dimaksudkan agar peneliti bisa mendapatkan gambaran mengenai permasalahan penjadwalan Staff rumah sakit yang ada di RSIA Kendangsari Surabaya.
	Dengan melakukan interview ini diharapkan peneliti mendapatkan informasi

Interview Protocol	
Human Resource Departemen view	
	mengenai penjadwalan staff rumah sakit yang sudah diterapkan Rumah Sakit Ibu dan Anak kendangsari Surabaya
	Untuk menjaga dan menjamin kerahasiaan, maka data – data yang bersifat pribadi akan dirahasiakan oleh peneliti.

Tabel A.2 Hasil wawancara (1)

NO	Soal	Jawaban
1	Berapa jumlah Staff yang dimiliki oleh RS	124 Staff (Keseluruhan medis dan non medis)
2	Ada berapakan Tipe staff staff rumah sakit dipekerjakan? (Skil Type)	Klasifikasi Staff rumah sakit dilakukan berdasarkan Pelatihan keterampilan yang dimiliki masing masing staff rumah sakit. Secara keseluruan staff rumah sakit yang dimiliki oleh RSIA khusunya perawat hanya memiliki sampai PK2 (Pelatihan Keterampilan tingkat 2) sedangkan untuk perawat senior atau kepala perawat memiliki PK3 (Pelatihan Keterampilan tingkat 3)
3	Berapa jumlah ward atau bangsal yang dimiliki oleh RS?	RSIA memiliki 14 unit atau bangsal, terdapat 6 unit RSIA yang memiliki pattern yang unik dalam menjadwalkan staff baik medis maupun non medis yaitu: • Kamar Operasi

NO	Soal	Jawaban
		• Ruang bayi dan NICU • SIM dan RM • Instalasi Farmasi • Instalasi rawat jalan dan IGD • Instalasi Gizi
4	Bagaimana cara pembuatan jadwal staff rumah sakit saat ini, apakah secara terpusat atau dibagi setiap ward atau bangsal?	Pembuatan Jadwal dilakukan secara per unit yang dilakukan oleh masing masing PJ unit. PJ unit mengumpulkan jadwal kepada bagian SDM RSIA maksimal tanggal 25 pada setiap bulan untuk bulan berikutnya untuk diinputkan ke dalam sistem mereka.
5	Bagaimanakah aturan umum mengenai penjadwalan staff rumah sakit?	• Staff yang memperoleh shift malam sebanyak maka harus memperoleh libur dihari berikutnya • Setiap staff dalam satu bulan harus mendapatkan bagian shift malam • Staff memiliki maksimal jam kerja 7 jam per hari dalam 6 hari perminggu • Jam lembur dihitung minimal 1 jam
6	Siapa yang melakukan penjadwalan staff rumah sakit?	Penjadwalan dilakukan oleh setiap pj unit atau bangsal

NO	Soal	Jawaban
7	Berapa kali penjadwalan staff rumah sakit dilakukan? Apakah setiap minggu, atau bulan?	Penjadwalan staff rumah sakit dilakukan sekali dalam sat bulan
8	Apakah ada rotasi perpindahan staff rumah sakit yang dilakukan?	Ada, akan tetapi perpindahan jadwal staff rumah sakit terseut biasanya tanpa sepengetahuan pihak SDM. Kebanyakan staff rumah sakit mengganti shift mereka sendiri tanpa melakukan konfirmasi terlebih dahulu kepada pihak SDM.
9	Berapa shift yang dijalankan setiap hari, dan berapa lama alokasi waktu setiap shift?	Secara umum terdapat 3 shift dengan pembagian 7 jam yaitu ➔ Pagi: 07.00-14.00 ➔ Sore: 14.00-21.00 ➔ Malam: 21.00-07.00 untuk mengatasi jam kerja yang sibuk pada jam jam tertentu RSIA menerapkan shift middle yang berlangsung dari jam 10.00-17.00 atau jam 12.00-19.00
10	Berapa maksimum jam kerja yang didapatkan masing masing staff rumah sakit dalam satu minggu?	Tidak ada maksimum jam kerja setiap minggunya. Yang ada hanyalah batas minimal libur untuk setiap bulannya. Setiap bulannya, jatah libur untuk setiap staff rumah sakit diusahakan harus sama. Apabila tidak sama, maka staff rumah sakit yang

NO	Soal	Jawaban
		mendapat libur lebih sedikit akan diganti hari liburnya di bulan depan.
11	Bagaimana penjadwalan staff rumah sakit yang sudah dilakukan saat ini? Apakah terdapat keluhan dari staff rumah sakit mengenai penjadwalan yang sudah ada?	Penjadwalan staff rumah sakit RSIA saat ini dilakukan secara menual oleh masing masing PJ bangsal atau unit RS. Hal tersbut mengakibatkan memakan banyak waktu.
12	Apa saja kekurangan penjadwalan yang telah diterapkan saat ini? apa yang perlu dioptimalkan misal di tingkat manajemen atau di tingkat staff rumah sakit?	Pihak SDM kesulitan untuk mengontrol jalanya penjadwalan dikarenakan banyak dari staff rumah sakit yang berganti shift kerja tanpa ada izin terlebih dahulu. Hal yang ingin dioptimalkan oleh RS adalah bagaimana cara mengontrol jalannya penjdawalan dengan menggunakan sistem yang langsung terintegrasi langsung dengan fingerprint yang ada sehinga kecurangan atau pelanggaran Staff rumah sakit akan saling tukar menukar jam dapat diminimalisasi. Masalah lain juga adanya jam jam dimana rumah sakit sepi atau sangat ramai. Ketika rumah sakit keadaannya sangat ramai, masalah

NO	Soal	Jawaban
		tersebut biasanya diselesaikan dengan cara mengoper staff rumah sakit yang bekerja pada shift tersebut pada unit yang renggang ke unit yang padat pasien. Namun apabila rumah sakit sepi, maka bagian SDM akan meliburkan staff rumah sakitnya. Misal dokter yang bertugas di RS tersebut harus keluar negeri semuanya untuk mengikuti pekan ilmiah. Melihat hal ini maka SDM berinisiatif untuk meliburkan staff rumah sakit yang berada pada shift dan unit tersebut
13	Bagaimana skil yang dimiliki staff rumah sakit apakah ada staff rumah sakit khusus yang menangani kasus tertentu (Type skil masing-masing staff rumah sakit)	Bagian unit yang harus memiliki staff rumah sakit skil khusus adalah: ➢ Ruang Operasi ➢ Rekam Medis
14	Dalam setiap bangsal, kompetensi/*skil* apa saja yang diperlukan?	Hal tersebut tentunya tergantung bangsal atau unit. Untuk perawat kita membutuhkan skil perawat dengan tingkatan PK 1 untuk perawat biasa. Untuk ketua perawat atau perawat ahli minimal kita membutuhkan perawat yang memiliki skil

NO	Soal	Jawaban
		pelatihan sampai tingkat PK 2 dan PK 3.
15	Apakah setiap ward memiliki tipe skil staff rumah sakit tertentu?	Tentunya, karena kita pastinya menyesuaikan kebutuhan dan Tupoksi masing masing unit yang ada di rumah skit kami.
16	Bagaimana jadwal yang sudah diatur, tiba-tiba mengalami perubahan karena salah staff rumah sakit meminta libur?	Jatah ambil cuti harus ditentukan masing masing staff rumah sakit pada saat PJ unit melakukan penjadwalan. Dalam satu bulan staff rumah sakit diberikan jatah cuti satu hari. Pengajuan cuti dilakukan staff rumah sakit maksimal tanggal 25 untuk setiap bulannya. Untuk kasus berbeda, misalkan terdapat keluarga meninggal atau terkena musibah yang lain berarti dapat dikurangi dengan jatah cuti yang sudah dia tentukan pada awal penjadwalan.
17	Apakah ada staff rumah sakit yang memiliki pengurangan jam kerja? (ex. staff rumah sakit yang sedang hamil, staff rumah sakit memiliki keterbatasan)	Tidak ada pengurangan jam kerja setiap shiftnya untuk staff rumah sakit yang hamil. Namun terdapat pemberian cuti selama 3 bulan yaitu 1.5 bulan sebelum melahirkan dan 1,5 bulan sesudah melahirkan. Pengambilan cuti ini bersifat fleksibel.

Tabel A.3 Interview Protocol (2)

Interview Protocol	
Human Resource Departemen view	
Informasi Interview	
Interviewer	Zuli Maulidati
Narasumber	Ibu Silvy
Hari tanggal	Jum'at, 24 November 2017
Pukul	15.30-17.00
Lokasi	RSIA Merr Kendnagsari
Informasi Narasumber	
Nama	Bu Sylvy Medtasya Dzykrzyanka, S. Farm, M. Farm. Klin, APT, MARS
Jabatan	Kepala Bagian
Divisi	Sumber Daya Manusia (SDM) & Hukum
Instansi	RSIA Merr Kendangsari
Lama Bekerja	3.5 tahun
Penjelasan Interview	Interview ini bertujuan untuk salah satu sumber data untuk tugas akhir dengan judul "Optimasi Penjadwalan Staff rumah sakit Dengan Menggunakan Algoritma *Tabu Search Based Hyper-Heuristics* (Studi Kasus: Rumah Sakit Ibu Dan Anak Kendangsari)" yang dimaksudkan agar peneliti bisa mendapatkan gambaran mengenai permasalahan penjadwalan Staff rumah sakit yang ada di RSIA Kendangsari Surabaya secara spesifik setiap unit atun instalasi rumah sakit yang akan dijadwalkan.
	Dengan melakukan interview ini diharapkan peneliti mendapatkan detail informasi mengenai penjadwalan staff unit Farmasi, IGD, Ruang bayi & NICU, SIM &

Interview Protocol	
Human Resource Departemen view	
	RM, dan Ruang Operasi pada Rumah Sakit Ibu dan Anak kendangsari Surabaya
	Untuk menjaga dan menjamin kerahasiaan, maka data – data yang bersifat pribadi akan dirahasiakan oleh peneliti.

Tabel A.4 Hasil wawancara (2)

NO	Soal	Jawaban
1	Apa itu jadwal lepas malam? Apa perbedaannya dengan libur? Mengapa tidak sama dengan libur?	Shift Lepas malam merupakan shift yang hanya boleh ada ketika pola shift mendapatkan dua kali malam berturut-turut. Shift lepas malam merupakan shift pengganti jam kerja shift malam yang melebihi 7 jam yaitu 10 jam setiap shift malam. Jadi shift malam dianggap masuk kerja dengan jam kerja 2 malam sebelumnya yaitu 6 jam kerja. Setelah itu staff mendapatkan jatah libur dari pagi sampai malam.
2	Mengapa ada pekerja dengan shift yang pagi terus?	Staff yang memperoleh shift pagi merupakan kepala unit staff yang berurusan langsung dengan pihak menejemen. Sehingga kehadiranya dibutuhkan setaip shift pagi yang bersamaan jam kantor bagian manajemen.

NO	Soal	Jawaban
3	Apakah perbedaan dari libur full dan libur lepas malam?	Libur Lepas malam bukan Libur tetapi pengganti jam kerja lebih pada 2 shif malam sebelumnya Libur: merupakan jatah kosong shift dari pagi sampai malam
4	Adakah keadilan di sana? Misal ada pekerja dengan shift pagi terus.	Untuk hal itu sudah kebutuhan dan peraturan kita, sehingga masing masing staf harus bisa menerima dengan legowo
5	Bagaimanakah aturan libur dalam rumah sakit?	Aturan libur ditentukan jumlah hari minggu dalam satu bulan dan libur nasional yang ada di bulan tersebut. Akan tetapi jatah libur tidak bisa diambil dalam waktu yang sama. Sehingga cara menjadwalkanya dengan cara mengikuti pola M-M-LM-L dari bulan sebelumnya. Untuk kepala unit memiliki hari libur setiap hari minggu seperti jadwal bagian kanto
6	Kenapa middle shift bisa ada 2 jenis? Identifikasi masing-masing. Misal apakah pekerja harus memiliki middle shift atau bagaimana.	Shift midle secara resmi dari rumah sakit merupakan rentan antara pukul 10.00-17.00. Sedangkan tipe middle shift 2 tergantung kebutuhan masing masing unit.
7	Jumlah jam kerjanya apakah harus sama?	Bisa langsung ditanyakan kepada unit yang bersangkutan

NO	Soal	Jawaban
8	Pembagian middle shiftnya bagaimana? Apakah tergantung dokter praktek?	Iya harus sama. Jumlah jam kerja setiap hari adalah 7 jam dalam 6 hari sehingaa dalam satu minggu setidaknya staff harus mendapatkan minimal 42 jam kerja
9	Bagaimana regulasi terkait penjadwalan? (pelajari dokumen dari kementrian ketenagakerjaan)	Penentuan pola middle dapat terjadi pada tanggal tanggal yang diprediksi banyak pasien, jadwal parktik dokter, dan sisa perseonel yang belum kebagian shift yang dijadwalkan
10	Dalam setahun berapa hari harus bekerja dan berarpa jam kerja?	
11	Bagaimana pola penjdwalan staff yang sudah diterapkan di unit ini?	Tidak ada batasan, hal tersebut kita hitung saja dengan hari libur yang ada.
12	Lihat polanya, apakah berulang untuk melakukan penjadwalan?	Pola umum yang harus ada adalah M-M-LM-L
13	Apakah pembuatan jadwal sudah ada standarisasi?	Belum terstandarisasi
14	Apakah tidak menimbulkan kecemburuan apabila jumlah shift malam tidak sama antar pekerjanya di setiap bulan?	Selama ini belum dapat keluhan tersebut. Harus legowo sih masing masing staff kita.

NO	Soal	Jawaban
15	Apakah ada hard constraint tertentu misal untuk pekerja yang bekerja di shift pagi terus? Misal dari skil yang dipunyai atau jabatan yang dimiliki	Hal tersebut langsung disesuaikan masing- masing kebutuhan unit. Akan tetepi jika staff yang selelu mendepatkan shift pagi merupakan kebala bagian. Hal tersebut dikarenakan kebutuhan dengan pihak menejemen agar lebih mudah misalkan rapat, atau kordinasi semacamnya.

Tabel A.5 Interview Protocol (3)

Interview Protocol	
Human Resource Departemen view	
Informasi Interview	
Interviewer	Zuli Maulidati
Narasumber	Tyas
Hari tanggal	Jum'at, 24 November 2017
Pukul	16.40-17.20
Lokasi	RSIA Merr Kendnagsari
Informasi Narasumber	
Nama	Ibu Tyas
Jabatan	Kepala Bagian dan Ahli Gizi
Divisi	Instalasi Gizi dan Cafe
Instansi	RSIA Merr Kendangsari
Lama Bekerja	2 tahun
Penjelasan Interview	Interview ini bertujuan untuk salah satu sumber data untuk tugas akhir dengan judul "Optimasi Penjadwalan Staff rumah sakit Dengan Menggunakan Algoritma *Tabu*

Interview Protocol	
Human Resource Departemen view	
	Search Based Hyper-Heuristics (Studi Kasus: Rumah Sakit Ibu Dan Anak Kendangsari)" yang dimaksudkan agar peneliti bisa mendapatkan gambaran mengenai permasalahan penjadwalan Staff rumah sakit yang ada di RSIA Kendangsari Surabaya secara spesifik setiap unit atun instalasi rumah sakit yang akan dijadwalkan.
	Dengan melakukan interview ini diharapkan peneliti mendapatkan detail informasi mengenai penjadwalan staff unit Gizi pada Rumah Sakit Ibu dan Anak kendangsari Surabaya
	Untuk menjaga dan menjamin kerahasiaan, maka data – data yang bersifat pribadi akan dirahasiakan oleh peneliti.

Tabel A.6 Hasil wawancara (3)

NO	Soal	Jawaban
1	Bagaimana pola penjdwalan staff yang sudah diterapkan di unit ini?	Pola penjadwalan unit gizi dan Café memiliki pola yang cukup unik. Hal tersebut menyesuaikan kebutuhan unit kami dan penyesuaian dengan sumber daya manusia yang kami miliki, dikarenakan unit gizi tidak hanya menyediakan makanan kepada pasien akan tetapi juga kepada staf dan juga membuka Café. Jika pola secara umum kami mengikuti pola yang ditentukan rumah sakit yaitu Pola umum yang harus ada adalah M-M-LM-L

NO	Soal	Jawaban
2	Ada berapakah Shift pada unit Gizi mengapa ada pembagian P1, p2 dan p pada jenis-jenis shift.	Pembagian p1, p2, dan p merupakan sama sama shift pagi akan tetapi jam usai dan jam selesainya berbeda, Secara umum spsifikasi shift pada instalasi gizi ada: ✓ P1 (04.00-11.30) ✓ P2 (05.00-12.00) ✓ P (06.00-13.00) ✓ PS (06.00-20.00) ✓ M (21.00-07.00) ✓ S (13.00-20.00) Pembagian shift tersebut dilakukan untuk memudahkan tata kerja unit gizi. Beberapa ketentuan penjadwalan unit gizi adalah: ✓ Shift malam hanya untuk penyaji saja ✓ Shift P1 dan P2 hanya untuk chef dan helper ✓ Shift PS merupakan shift yang diperuntukan sopir yang bertugas mengakomodasikan unit gizi & Café bukan ambulan ✓ Shift P1 hanya untuk diperumtukan staf laki laki diakrenakan jam mulainyayang terlalu pagi ✓ Shift pagi terdiri dari 1 chef dan 1 helper. ✓ Shift P1 biasanya diisi helper laki-laki. ✓ Shift middle tidak ada helper, hanya ada chef
3	Bagaimana cara penyusunan	Penyusunan jadwal: pertama masukkan dulu permintaan libur dan

NO	Soal	Jawaban
	jadwal? Apakah berulang untuk melakukan penjadwalan?	cuti dari setiap staf. Staf juga bisa meminta shift pagi/siang. Setelah itu baru disusun jadwal untuk keseluruhannya. Pola umum yang harus ada adalah M-M-LM-L
4	Bagaimana menetukan Libur atau Cuti?	Permintaan libur dilakukan berdasarkan jumlah hari minggu dalam satu bulan, akan tetapi hari libur setiap staf tidak semua jatuh di hari minggu. Untuk penetapan cuti, staf dapat mengajukan permintaan cuti, sedangkan staf yang tidak mengambil jatah cuti nya dapat dijadikan sebagai kerja lembur.
5	Apakah tidak menimbulkan kecemburuan apabila jumlah shift malam tidak sama antar pekerjanya di setiap bulan?	Selama ini belum dapat keluhan tersebut. Harus legowo sih masing masing staff kita.
6	Bagaimana penentapan Shift middle pada unit Gizi?	Kapan harus ada middle? Ketika semua jadwal sudah terpenuhi (libur, cuti, malam). Kalau ada dokter yang praktek, maka kemungkinan bagian chef sama cafe memiliki shift middle.

LAMPIRAN B: Hasil Generate Jadwal

1. Hasil generate jadwal unit Farmasi

Pada Tabel B.1 merupakan hasil pembentukan jadwal secara otomatis berdasarkan pola pada jadwal manual bulan November 2017. Jadwal otomatis dihasilkan selema 31 hari di bulan desmber 2017. Kolom id merupakan nomer identitas staf yang dijadwalkan. Pada pengalokasian shift terdapat keterangan hari dan tanggal. Dimana J adalah jum'at, Sb adalah sabtu, M adalah minggu, Sn adalah senin, Sl adalah selesa, R adalah rabo, dan K adalah kamis.

Staf ber id satu memiliki pola selalu pagi, staf ber id dua sampai delapan memiliki pola malam-malam lepas malam, kemudian libur. dan staf ber id dua memiliki pola selalu siang. Pengsisan middle shift apabila memiliki jumalah staf yang lebih pada hari itu.

Tabel B.1 Hasil generate jadwal unit Farmasi

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L
2	S	S	L	S	S	S	S	S	S	L	S	S	S	S	S	S	L	S	S	S	S	S	S	L	S	S	S	S	S	S	L

3	M	M	LM	L	P	P	S	S	MS	S	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M
4	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	S
5	S	S	S	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	S	M	M	LM	L	P	P	S
6	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P
7	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	S	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM
8	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L

2. Hasil generate jadwal unit Nicu dan Ruang Bayi

Pada Tabel B.2 merupakan hasil pembentukan jadwal secara otomatis berdasarkan pola pada jadwal manual bulan November 2017. Jadwal otomatis dihasilkan selema 31 hari di bulan desmber 2017. Kolom id merupakan nomer identitas staf yang dijadwalkan. Pada pengalokasian shift terdapat keterangan hari dan tanggal. Dimana J adalah jum'at, Sb adalah sabtu, M adalah minggu, Sn adalah senin, Sl adalah selesa, R adalah rabo, dan K adalah kamis.

Staf ber id satu memiliki pola selalu pagi, staf ber id dua sampai tiga belas memiliki pola malam-malam lepas malam, kemudian libur. Pengsisan middle shift apabila memiliki jumalah staf yang lebih pada hari itu.

Tabel B.2 Hasil generate Jadwal Unit Nicu dan Ruang bayi

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L
2	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M
3	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS
4	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S
5	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P
6	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM
7	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M

8	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS
9	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S
10	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P
11	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM
12	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M
13	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	MS

3. Hasil generate jadwal Ruang Operasi

Pada Tabel B.3 merupakan hasil pembentukan jadwal secara otomatis berdasarkan pola pada jadwal manual bulan November 2017. Jadwal otomatis dihasilkan selema 31 hari di bulan desmber 2017. Kolom id merupakan nomer identitas staf yang dijadwalkan. Pada pengalokasian shift terdapat keterangan hari dan tanggal. Dimana J adalah jum'at, Sb adalah sabtu, M adalah minggu, Sn adalah senin, Sl adalah selesa, R adalah rabo, dan K adalah kamis.

Seluruh staf di ruang operasi memiliki pola pembagian shift yang sama yaitu pagi dan siang. Untuk setiap hari minggu penjadwalan harus memiliki empat staf pada shift pagi dan dua staf lainya diberikan libur.

Tabel B.3 Hasil generate Jadwal Ruang Operasi

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	L	P	S	L	P	P	S	P	P	P	S	P	S	L	P	P	S	S	P	P	S	L	S	L	P	P	S	S	P
2	S	P	L	S	P	S	L	P	P	P	S	P	P	S	P	S	P	P	P	S	S	P	P	L	P	S	L	P	P	S	P
3	S	S	P	P	S	P	S	L	P	P	S	S	P	P	S	P	L	L	P	P	S	S	P	P	S	P	S	L	P	P	P
4	P	S	P	P	P	S	P	S	L	P	P	S	P	P	P	S	L	S	L	P	P	S	P	P	P	S	P	S	L	P	P
5	P	P	P	P	P	P	S	P	S	L	P	P	S	P	P	P	P	P	S	L	P	P	S	P	P	P	S	P	S	L	L
6	L	P	P	S	P	P	P	S	P	L	L	P	P	S	P	P	P	S	P	S	L	P	P	P	P	P	P	S	P	S	L

4. Hasil generate jadwal unit SIM dan RM

Pada Tabel B.4 merupakan hasil pembentukan jadwal secara otomatis berdasarkan pola pada jadwal manual bulan November 2017. Jadwal otomatis dihasilkan selema 31 satu di bulan desmber 2017. Kolom id merupakan nomer identitas staf yang dijadwalkan. Pada pengalokasian shift terdapat keterangan hari dan tanggal. Dimana J adalah jum'at, Sb adalah sabtu, M adalah minggu, Sn adalah senin, Sl adalah selesa, R adalah rabo, dan K adalah kamis.

Seluruh staf di unit SIM dan RM memiliki pola malam dua kali, lepas malam, libur, kemudian pagi dua kali, siang dua kali. Pengsisan *middle shift* apabila memiliki jumalah staf yang lebih pada hari itu.

Tabel B.4 Hasil generate Jadwal Unit SIM dan RM

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M
2	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS
3	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S

4	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P
5	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P
6	S	S	P	MS	M	M	LM	L	P	P	P	L	S	S	P	MS	M	M	LM	L	P	P	P	L	S	S	P	MS	M	M	LM

5. Hasil generate jadwal unit IGD dan Rawat Jalan

Pada Tabel B.5 merupakan hasil pembentukan jadwal secara otomatis berdasarkan pola pada jadwal manual bulan November 2017. Jadwal otomatis dihasilkan selema 31 hari di bulan desmber 2017. Kolom id merupakan nomer identitas staf yang dijadwalkan. Pada pengalokasian shift terdapat keterangan hari dan tanggal. Dimana J adalah jum'at, Sb adalah sabtu, M adalah minggu, Sn adalah senin, Sl adalah selesa, R adalah rabo, dan K adalah kamis.

Seluruh staf di unit IGD dan rawat jalan memiliki pola malam dua kali, lepas malam, libur, kemudian pagi dua kali, siang dua kali. Pengsisan *middle shift* apabila memiliki jumalah staf yang lebih pada hari itu.

Tabel B.5 Hasil generate jadwal Uinit IGD dan Rawat Jalan

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M

1	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M
2	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS
3	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L
4	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S
5	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S
6	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P
7	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L
8	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M

6. Hasil generate jadwal Unit Gizi dan Cafe

Pada Tabel C.4 merupakan hasil pembentukan jadwal secara otomatis berdasarkan pola pada jadwal manual bulan November 2017. Jadwal otomatis dihasilkan selema 31 hari di bulan desmber 2017. Kolom id

merupakan nomer identitas staf yang dijadwalkan. Pada pengalokasian shift terdapat keterangan hari dan tanggal. Dimana J adalah jum'at, Sb adalah sabtu, M adalah minggu, Sn adalah senin, Sl adalah selesa, R adalah rabo, dan K adalah kamis.

Tabel B.6 Hasil generate jadwal Unit Gizi dan Cafe

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L	P	P	P	P	P	P	L
2	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1
3	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD
4	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S
5	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P
6	S	S	S	P1	P1	L	P1	P1	S	S	S	P1	P1	L	P1	P1	S	S	S	P1	P1	L	P1	P1	S	S	S	P1	P1	L	P1

7	P2	P	S	S	L	P	S	P2	P	MS	P2	P	S	S	L	P	S	P2	P	MS	P2	P	S	S	L	P	S	P2	P	MS	P2
8	P	MS	P2	P	S	S	L	P	S	P2	P	MS	P2	P	S	S	L	P	S	P2	P	MS	P2	P	S	S	L	P	S	P2	P
9	S	P2	P	MS	P2	P	S	S	L	P	S	P2	P	MS	P2	P	S	S	L	P	S	P2	P	MS	P2	P	S	S	L	P	S
10	S	S	P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P
11	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M
12	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P
13	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S
14	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P
15	LM	L	P	P	S	S	MD	MD	M	M	LM	L	P	P	S	S	MD	MD	M	M	LM	L	P	P	S	S	P	MD	M	M	LM
16	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L

17	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS
18	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS
19	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L

LAMPIRAN C: Uji Coba Optimasi

Tabel C.1 Uji coba optimasi jadwal Unit Farmasi

	Tabu Search						Hill Climbing			
	Jumlah Iterasi		Nilai			Jumlah Iterasi		Nilai	
NO	100 kali	200 kali			NO	100 kali	200 kali		
1	0.9223	0.9301	Max	0.976879	1	0.84668	0.9301	Max	0.960908
2	0.882	0.9192	Min	0.856164	2	0.841184	0.9055	Min	0.82021
3	0.9769	0.9438	Avg	0.905252	3	0.941265	0.904	Avg	0.882481
4	0.8562	0.9069			4	0.890051	0.9441		
5	0.9343	0.9534	Max	0.968478	5	0.91385	0.9334	Max	0.957274
6	0.9086	0.9011	Min	0.803342	6	0.94186	0.9537	Min	0.859697
7	0.8766	0.8033	Avg	0.905989	7	0.960908	0.9232	Avg	0.911613
8	0.9032	0.9265			8	0.886679	0.9128		
9	0.8795	0.9109			9	0.91125	0.9573		
10	0.9595	0.8525			10	0.900133	0.8752		

NO	Tabu Search					NO	Hill Climbing				
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai		
	100 kali	200 kali					100 kali	200 kali			
11	0.8799	0.9171				11	0.904381	0.8836			
12	0.9172	0.8911				12	0.82021	0.8903			
13	0.9098	0.8999				13	0.824519	0.8662			
14	0.8895	0.9685				14	0.845579	0.898			
15	0.9165	0.8274				15	0.917137	0.9331			
16	0.9522	0.9575				16	0.84668	0.9206			
17	0.884	0.8853				17	0.82113	0.937			
18	0.8579	0.9408				18	0.901872	0.8823			
19	0.9267	0.8467				19	0.875172	0.9026			
20	0.9055	0.9042				20	0.886119	0.9311			
21	0.8723	0.9398				21	0.855447	0.8597			

Tabel C.2 Uji coba optimasi jadwal Unit Nicu dan Ruang Bayi

NO	Tabu Search				NO	Hill Climbing			
	Jumlah Iterasi		Nilai			Jumlah Iterasi		Nilai	
	100 kali	200 kali				100 kali	200 kali		
1	0.936641	0.946371	Max	0.986343	1	0.939056	0.938269	Max	0.98263
2	0.933553	0.965784	Min	0.868682	2	0.900487	0.970232	Min	0.871989
3	0.919914	0.918935	Avg	0.930831	3	0.929478	0.92063	Avg	0.932436
4	0.961041	0.942177			4	0.954338	0.930417		
5	0.911238	0.955136	Max	0.986343	5	0.98263	0.949093	Max	0.970232
6	0.92562	0.904977	Min	0.868682	6	0.899245	0.902492	Min	0.891277
7	0.923325	0.898654	Avg	0.932668	7	0.955387	0.938583	Avg	0.927069
8	0.933048	0.914814			8	0.871989	0.905132		
9	0.870794	0.868682			9	0.892578	0.93474		
10	0.972672	0.889611			10	0.956231	0.915577		
11	0.892126	0.941774			11	0.964459	0.897308		
12	0.936502	0.954087			12	0.966711	0.911381		
13	0.948322	0.967715			13	0.927156	0.925782		

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 kali	200 kali					100 kali	200 kali		
14	0.914496	0.986343				14	0.946347	0.918372		
15	0.942029	0.917662				15	0.910828	0.940856		
16	0.935753	0.963167				16	0.913753	0.928191		
17	0.949867	0.890298				17	0.942177	0.918079		
18	0.9316	0.949451				18	0.94599	0.891277		
19	0.927672	0.937785				19	0.947557	0.942323		
20	0.93765	0.944056				20	0.935511	0.922911		
21	0.943586	0.928541				21	0.899239	0.96681		

Tabel C.3 Uji coba optimasi jadwal Unit IGD dan Rawat Jalan

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
1	0.935233	0.945896	Max	0.960227		1	0.914179	0.945896	Max	0.96305

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
2	0.909605	0.91967	Min	0.897989		2	0.950415	0.91967	Min	0.888889
3	0.960133	0.946248	Avg	0.932587		3	0.961376	0.946248	Avg	0.928116
4	0.941176	0.931338				4	0.961219	0.931338		
5	0.916265	0.925714	Max	0.978827		5	0.96305	0.925714	Max	0.978827
6	0.926383	0.933519	Min	0.888889		6	0.914557	0.933519	Min	0.888889
7	0.927507	0.909441	Avg	0.928741		7	0.939083	0.909441	Avg	0.928741
8	0.907518	0.942185				8	0.92803	0.942185		
9	0.960227	0.923077				9	0.913242	0.923077		
10	0.933689	0.903583				10	0.924423	0.903583		
11	0.920652	0.941213				11	0.888889	0.941213		
12	0.933519	0.928163				12	0.914557	0.928163		
13	0.939417	0.902778				13	0.956006	0.902778		
14	0.935275	0.978827				14	0.927336	0.978827		
15	0.948142	0.938312				15	0.896534	0.938312		

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
16	0.956633	0.914557				16	0.946145	0.914557		
17	0.936768	0.920918				17	0.928367	0.920918		
18	0.897989	0.888889				18	0.925641	0.888889		
19	0.916265	0.917576				19	0.922945	0.917576		
20	0.943146	0.932561				20	0.911672	0.932561		
21	0.938793	0.959094				21	0.902778	0.959094		

Tabel C.4 Uji coba optimasi jadwal Unit SIM dan RM

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
1	0.98321	0.995902	Max	1		1	0.988066	0.991991	Max	0.998223
2	0.991991	0.986513	Min	0.979592		2	0.989945	0.99681	Min	0.979592
3	0.979592	0.988066	Avg	0.986782		3	0.989691	0.979592	Avg	0.986564

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
4	0.991991	0.992658				4	0.982704	0.982912		
5	0.98321	0.998223	Max	0.998223		5	0.983626	0.988683	Max	0.998223
6	0.988034	0.982704	Min	0.982704		6	0.992969	0.979592	Min	0.979592
7	0.98321	0.98321	Avg	0.989366		7	0.98321	0.979592	Avg	0.985206
8	0.979592	0.997455				8	0.982993	0.979592		
9	0.982704	0.988304				9	0.998223	0.998223		
10	0.97971	0.991991				10	0.985507	0.979592		
11	0.982704	0.98321				11	0.983988	0.985614		
12	0.982993	0.996552				12	0.989782	0.98321		
13	0.988066	0.992658				13	0.987805	0.99681		
14	1	0.982704				14	0.98321	0.979592		
15	0.99409	0.993663				15	0.983806	0.989782		
16	0.980392	0.991837				16	0.985614	0.982704		
17	0.985614	0.992658				17	0.982704	0.982704		

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
18	0.995452	0.98321				18	0.987805	0.98321		
19	0.99835	0.98321				19	0.988304	0.982704		
20	0.98321	0.988359				20	0.979592	0.98321		
21	0.988304	0.983607				21	0.988304	0.98321		

Tabel C.5 Uji coba optimasi jadwal Unit Gizi dan Cafe

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
1	0.837821	0.842263	Max	0.861071		1	0.806431	0.823053	Max	0.857769
2	0.767918	0.773158	Min	0.743789		2	0.784029	0.750548	Min	0.714176
3	0.811861	0.763472	Avg	0.788966		3	0.758716	0.799674	Avg	0.786249
4	0.748928	0.775861				4	0.800363	0.852278		
5	0.861071	0.860951	Max	0.860951		5	0.857769	0.818802	Max	0.861071

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
6	0.791113	0.853957	Min	0.743789		6	0.752691	0.799238	Min	0.72945
7	0.822503	0.766517	Avg	0.792502		7	0.808282	0.861071	Avg	0.800686
8	0.783142	0.812004				8	0.742142	0.819983		
9	0.766902	0.752691				9	0.747924	0.752263		
10	0.770062	0.85056				10	0.743789	0.803659		
11	0.795882	0.762632				11	0.823711	0.80021		
12	0.836877	0.780001				12	0.773733	0.806431		
13	0.80245	0.793419				13	0.753158	0.736361		
14	0.788956	0.744819				14	0.838953	0.781499		
15	0.754314	0.760217				15	0.792299	0.813063		
16	0.750223	0.841491				16	0.759242	0.828049		
17	0.769588	0.771135				17	0.79477	0.83811		
18	0.813006	0.743789				18	0.827139	0.72945		
19	0.743789	0.764839				19	0.823459	0.842721		

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	100 Kali	200 Kali					100 Kali	200 Kali		
20	0.781264	0.77041				20	0.808445	0.743789		
21	0.770616	0.858352				21	0.714176	0.814147		

Tabel C.6 Uji coba optimasi jadwal Ruang Operasi

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	50 kali	100 Kali					50 kali	100 Kali		
1	0.968559	0.968559	Max	0.988359		1	0.968559	0.968559	Max	0.981669
2	0.968559	0.978359	Min	0.968559		2	0.981669	0.968559	Min	0.968559
3	0.968559	0.968559	Avg	0.971688		3	0.981669	0.968559	Avg	0.970896
4	0.968559	0.968559				4	0.981669	0.968559		
5	0.968559	0.968559	Max	0.981669		5	0.968559	0.968559	Max	0.981669
6	0.981669	0.981669	Min	0.968559		6	0.968559	0.968559	Min	0.968559
7	0.971803	0.968559	Avg	0.970895		7	0.968559	0.981669	Avg	0.971056

NO	Tabu Search					NO	Hill Climbing			
	Jumlah Iterasi		Nilai				Jumlah Iterasi		Nilai	
	50 kali	100 Kali					50 kali	100 Kali		
8	0.968559	0.968559				8	0.968559	0.981669		
9	0.968559	0.968559				9	0.968559	0.968559		
10	0.981669	0.968559				10	0.968559	0.981669		
11	0.968559	0.978359				11	0.968559	0.968559		
12	0.968559	0.968559				12	0.968559	0.968559		
13	0.988359	0.968559				13	0.968559	0.968559		
14	0.968559	0.968559				14	0.971803	0.968559		
15	0.968559	0.981669				15	0.97507	0.968559		
16	0.968559	0.968559				16	0.968559	0.968559		
17	0.985003	0.968559				17	0.968559	0.968559		
18	0.968559	0.968559				18	0.968559	0.981669		
19	0.968559	0.971803				19	0.968559	0.968559		
20	0.968559	0.968559				20	0.968559	0.968559		
21	0.968559	0.968559				21	0.968559	0.968559		

LAMPIRAN D: Hasil Optimasi Penjadwalan

Pada Tabel D.1, Tabel D.2, Tabel D.3, Tabel D.4, Tabel D.5, dan, Tabel D.6 merupakan hasil optimasi jadwal pembentukan secara otomatis. Jadwal hasil optimasi berjumlah selema 31 hari di bulan desmber 2017. Kolom id merupakan nomer identitas staf yang dijadwalkan. Pada pengalokasian shift terdapat keterangan hari dan tanggal. Dimana J adalah jum'at, Sb adalah sabtu, M adalah minggu, Sn adalah senin, Sl adalah selesa, R adalah rabo, dan K adalah kamis.

Tabel D.1 Hasil Optimasi Jadwal Unit Farmasi

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	L	P	P	P	P	P	P	L	P	L	P	P	P	P	L	P	P	P	P	P	P	S	P	P	P	P	P	P	S
2	S	S	L	S	S	S	S	L	S	S	S	S	S	S	S	S	M	L	S	S	S	L	S	L	S	S	S	L	S	S	L
3	M	M	LM	S	P	P	S	S	MS	L	M	M	LM	L	P	P	S	S	L	L	M	M	LM	L	P	P	S	S	MS	MS	M

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
4	L	P	L	M	L	MS	P	P	S	S	P	L	M	M	LM	L	P	L	S	P	MS	P	M	M	LM	L	P	S	S	S	L
5	S	S	S	L	M	M	LM	S	P	L	S	S	P	S	M	M	LM	P	P	P	S	S	L	L	M	M	LM	L	P	P	S
6	P	P	S	L	S	L	M	M	LM	M	L	P	S	L	L	MS	L	M	LM	S	P	P	S	S	MS	MS	M	M	LM	L	P
7	LM	L	P	P	LM	S	L	L	M	L	LM	P	L	P	S	S	S	P	M	M	LM	S	P	P	S	S	L	P	M	M	LM
8	P	L	M	P	P	P	P	P	P	P	P	P	P	P	P	P	L	S	P	L	P	L	P	L	P	P	P	P	P	P	L

Tabel D.2 Hasil Optimasi Jadwal Unit Nicu dan Ruang Bayi

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	LM	P	P	P	P	P	P	S	P	L	P	P	P	P	L	P	P	P	P	P	P	P	P	P	L	P	P	P	L
2	M	M	L	L	P	P	S	S	L	MS	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	P	MS	M
3	L	MS	M	M	LM	M	LM	P	S	L	P	L	M	M	LM	P	P	P	S	S	L	MS	M	M	LM	S	P	P	S	S	L
4	L	S	L	MS	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	S	P	L	S	S	MS	MS	M	M	LM	L	L	P	S
5	P	P	S	L	L	L	M	L	LM	S	L	P	S	S	MS	MS	M	M	LM	S	P	P	S	S	MS	L	M	L	LM	L	P
6	LM	L	P	P	S	S	P	M	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	L	P	L	S	L	MS	M	M	M	LM
7	M	M	LM	L	P	P	S	S	L	MS	M	M	LM	L	P	L	M	L	L	L	M	M	LM	L	P	P	S	S	MS	MS	M

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
8	MS	L	M	M	LM	L	P	P	S	L	L	MS	M	M	LM	L	P	P	S	P	MS	L	M	M	LM	L	P	P	S	S	MS
9	S	S	L	S	M	L	L	L	P	P	S	S	MS	MS	M	M	LM	L	P	L	S	L	MS	P	M	L	LM	L	P	P	S
10	P	P	S	S	MS	L	M	M	LM	L	P	P	S	S	MS	L	S	M	LM	P	P	P	S	S	MS	L	M	M	LM	L	P
11	LM	L	P	MS	S	S	MS	L	M	M	LM	P	P	P	S	S	L	L	M	M	LM	S	P	L	S	S	P	L	M	M	LM
12	M	M	LM	P	P	P	S	S	L	L	M	M	LM	L	P	P	S	S	MS	MS	M	M	LM	L	P	P	S	S	L	MS	M
13	S	MS	M	M	LM	L	P	P	S	S	L	MS	M	M	LM	L	P	P	S	S	MS	L	M	M	LM	M	P	P	S	S	L

Tabel D.3 Hasil Optimasi Jadwal Unit SIM dan RM

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	L	S	S	MS	P	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	L	LM	L	P	P	P	P	S	P	MS	MS	M
2	P	P	P	L	S	S	MS	L	M	M	LM	L	P	P	P	P	S	S	MS	MS	M	M	LM	L	P	L	P	L	S	S	MS
3	LM	M	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	L	P	L	S	S	MS	MS	M	M	LM	L	P	L	P	L	S
4	M	L	LM	L	P	L	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	M	S	S	MS	MS	M	M	LM	L	P	P	P
5	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P	P	P	L	S	S	MS	MS	M	M	LM	L	P
6	S	S	P	L	M	M	LM	L	P	P	P	L	S	S	P	L	M	M	LM	L	P	P	P	L	S	S	P	S	M	M	LM

Tabel D.4 Hasil Optimasi Jadwal Unit IGD dan Rawat Jalan

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	M	M	LM	L	P	P	S	S	S	S	S	L	MD	L	S	M	M	LM	MD	P	P	S	S	L	S	S	L	MD	L	S	M
2	MS	S	M	M	LM	L	P	L	S	S	S	S	S	L	MD	L	S	M	M	L	L	P	P	S	S	S	S	S	L	MD	MS
3	L	MD	MS	S	M	M	LM	P	P	P	S	S	S	S	S	L	MD	L	S	M	M	LM	L	P	P	S	S	S	S	S	L
4	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	L	LM	S	M	M	LM	L	P	P	S	S	S	S
5	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	L	S	M	M	LM	L	P	P	S	S
6	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	L	MD	L	S	M	M	LM	L	P	P
7	L	P	P	S	S	S	S	S	L	MD	MS	S	M	M	LM	L	P	P	S	S	S	S	S	S	MD	MS	S	M	M	LM	L

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
8	M	LM	L	P	P	S	S	S	S	S	L	MD	L	S	M	M	LM	MS	P	P	S	S	S	S	S	L	MD	L	S	M	M

Tabel D.5 Hasil Optimasi Penjadwalan Unit Gizi dan Cafe

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	L	P	P	P	P	P	P	S	P	P	P	P	P	P	L	P	P	P	P	P	P	P	P	P	P	P	P	P	L
2	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	L	P	L	S	S	MD	MD	P1	L	S	P1	P	L	S	S	MD	MD	L
3	MD	MD	P1	P	S	P1	P	L	S	L	MD	MD	P1	P	S	P1	P	S	S	S	MD	MD	P1	P	S	P1	P	P2	S	S	MD
4	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
5	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P	L	S	S	MD	MD	P1	P	S	P1	P
6	S	S	S	P1	P1	L	L	P1	S	S	S	P1	P1	L	P1	L	S	S	S	L	P1	L	P1	P1	S	S	S	P1	P1	L	P1
7	P2	P	S	S	L	P	S	P2	P	L	P2	P	S	S	L	P	S	P2	P	P2	P2	P	S	S	L	P	S	L	P	L	P2
8	P	L	P2	P	S	L	S	P	S	P2	P	L	P2	P	S	S	L	P	S	P1	P	P	P2	P	S	S	L	P	S	P2	P
9	S	P2	P	L	P2	P	L	S	L	P	S	P2	P	L	P2	P	S	L	L	P	S	P2	P	L	P2	P	S	S	L	P	S
10	S	S	P	P	MD	MD	P1	S	S	P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P	P	MD	MD	L	S	S	P
11	M	M	L	L	P	P	S	S	P	P	M	M	LM	P1	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
12	P	P	M	M	LM	S	P	P	S	S	P	P	M	M	LM	P1	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P
13	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	L	M	M	LM	MD	P	P	S
14	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P	P	S	S	P	P	M	M	LM	L	P
15	LM	L	P	P	S	S	MD	MD	M	M	LM	L	P	P	S	S	MD	MD	M	M	LM	L	P	P	L	S	P	L	M	M	LM
16	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	S	P	P	S	S	S	P1
17	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS
18	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
19	S	S	LM	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L	PS	S	S	P	P	P	PS	L	P	P	S	S	S	L

Tabel D.6 Hasil Optimasi Jadwal Ruang Operasi

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
1	P	P	L	P	S	L	P	P	S	P	P	P	S	P	S	P	P	P	S	S	P	P	S	L	S	L	P	P	S	S	P
2	S	P	L	S	P	S	L	P	P	P	S	P	P	S	P	S	P	P	P	S	S	P	P	L	P	S	L	P	P	S	P
3	S	S	P	P	S	P	S	L	P	P	S	S	P	P	S	L	L	L	P	P	S	S	P	P	S	P	S	L	P	P	P
4	P	S	P	P	P	S	P	S	L	P	P	S	P	P	P	S	L	S	L	P	P	S	P	P	P	S	P	S	L	P	P
5	P	P	P	P	P	P	S	P	S	L	P	P	S	P	P	P	P	P	S	L	P	P	S	P	P	P	S	P	S	L	L

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M	Sn	Sl	R	K	J	Sb	M
6	L	P	P	S	P	P	P	S	P	L	L	P	P	S	P	P	P	S	P	S	L	P	P	P	P	P	P	S	P	S	L